Prompt 1:

"""

You are a helpful assistant. Use the following context to answer the question.

Context:

{context}

Question:

{question}

""")

question 1:

WHat is software enginerring?

1)

Model: meta-llama/Llama-3.3-70B-Instruct-Turbo-Free

Prompt : prompt 1

Question : question 1

Embedding model: sentence-transformers/all-MiniLM-L6

Chunking method: recursive text splitting chunk size 800 and overlap 100

Answer:

According to the provided context, software engineering is defined as:

\* "A process, methods for managing and engineering software, and tools" (Page 13)

\* "A discipline or a body of knowledge, [but] engineering is a verb, an action word, a way of approaching a problem" (Page 13, quoting Scott Whitmir)

\* "Encompass[ing] a process, methods, and tools that enable complex computer-based systems to be built in a timely manner with quality" (Page 53)

In general, software engineering is a field that encompasses a set of principles, methods, and tools used to design, develop, and maintain software systems. It involves a systematic approach to software development, focusing on quality, timeliness, and customer satisfaction.

📚 Context Chunks Used:

--- Chunk 1 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 53, Chunk software\_engineering.pdf:53:2]

Software engineering encompasses process, methods, and tools that enable complex computer-based systems to be built in a timely manner with quality. The software process incorporates five framework activities—communication, planning, modeling, construction, and deployment—that are applicable to all software proj- ects. Software engineering practice is a problem solving activity that follows a set of core principles. A wide array of software myths continue to lead managers and practitioners astray, even as our collective knowledge of software and the technologies required to build it grows. As you learn more about software engineering, you’ll begin to un- derstand why these myths should be debunked whenever they are encountered. PROBLEMS AND POINTS TO PONDER

--- Chunk 2 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 41, Chunk software\_engineering.pdf:41:4]

CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 13 uote: “More than a discipline or a body of knowledge, engineering is a verb, an action word, a way of approaching a problem.” Scott Whitmir How do we define software engineering? ? 9 For numerous additional definitions of software engineering, see www.answers.com/topic/ software-engineering#wp-\_note-13. 10 Quality management and related approaches are discussed in Chapter 14 and throughout Part 3 of this book. Software engineering encompasses a process, methods for managing and engineering software, and tools. pre75977\_ch01.qxd 11/27/08 3:11 PM Page 13

--- Chunk 3 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 25, Chunk software\_engineering.pdf:25:1]

It has been almost three decades since the first edition of this book was written. During that time, software engineering has evolved from an obscure idea practiced by a relatively small number of zealots to a legitimate engineering discipline. Today, it is recognized as a subject worthy of serious research, conscientious study, and tumultuous debate. Through- out the industry, software engineer has replaced programmer as the job title of preference. Software process models, software engineering methods, and software tools have been adopted successfully across a broad spectrum of industry segments. Although managers and practitioners alike recognize the need for a more disciplined approach to software, they continue to debate the manner in which discipline is to be

--- Chunk 4 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 25, Chunk software\_engineering.pdf:25:2]

approach to software, they continue to debate the manner in which discipline is to be applied. Many individuals and companies still develop software haphazardly, even as they build systems to service today’s most advanced technologies. Many professionals and students are unaware of modern methods. And as a result, the quality of the software that we produce suffers, and bad things happen. In addition, debate and controversy about the true nature of the software engineering approach continue. The status of software engi- neering is a study in contrasts. Attitudes have changed, progress has been made, but much remains to be done before the discipline reaches full maturity. The seventh edition of Software Engineering: A Practitioner’s Approach is intended to

--- Chunk 5 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 42, Chunk software\_engineering.pdf:42:1]

each area of the technology and include modeling activities and other descriptive techniques. Software engineering tools provide automated or semiautomated support for the process and the methods. When tools are integrated so that information created by one tool can be used by another, a system for the support of software development, called computer-aided software engineering, is established. 1.4 T HE SOFTWARE PROCESS A process is a collection of activities, actions, and tasks that are performed when some work product is to be created. An activity strives to achieve a broad objective (e.g., communication with stakeholders) and is applied regardless of the application domain, size of the project, complexity of the effort, or degree of rigor with which

2)

Model: meta-llama/Llama-3.3-70B-Instruct-Turbo-Free

Prompt : prompt 1

Question : question 1

Embedding model: model\_name="Qwen/Qwen3-Embedding-0.6B",

Chunking method: recursive text splitting chunk size 800 and overlap 100

📘 Answer:

Software engineering encompasses a process, methods, and tools that enable complex computer-based systems to be built in a timely manner with quality. It is a broad array of principles, concepts, methods, and tools that are used to plan and develop software. Software engineering is a problem-solving activity that follows a set of core principles and provides a road map for getting to a successful destination in the creation of a computer-based system or product. It involves a set of framework activities, including communication, planning, modeling, construction, and deployment, and is conducted by software engineers and their managers. Overall, software engineering is a discipline that aims to improve the ease with which changes can be accommodated and reduce the amount of effort expended when changes are made to software systems.

📚 Context Chunks Used:

--- Chunk 1 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 53, Chunk software\_engineering.pdf:53:2]

Software engineering encompasses process, methods, and tools that enable complex computer-based systems to be built in a timely manner with quality. The software process incorporates five framework activities—communication, planning, modeling, construction, and deployment—that are applicable to all software proj- ects. Software engineering practice is a problem solving activity that follows a set of core principles. A wide array of software myths continue to lead managers and practitioners astray, even as our collective knowledge of software and the technologies required to build it grows. As you learn more about software engineering, you’ll begin to un- derstand why these myths should be debunked whenever they are encountered. PROBLEMS AND POINTS TO PONDER

--- Chunk 2 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 124, Chunk software\_engineering.pdf:124:2]

requirements . .107 testing . . . . . . .112 What is it? Software engineering practice is a broad array of princi- ples, concepts, methods, and tools that you must consider as software is planned and developed. Principles that guide practice establish a foundation from which soft- ware engineering is conducted. Who does it? Practitioners (software engineers) and their managers conduct a variety of soft- ware engineering tasks. Why is it important? The software process pro- vides everyone involved in the creation of a computer-based system or product with a road map for getting to a successful destination. Practice provides you with the detail you’ll need to drive along the road. It tells you where the bridges, the roadblocks, and the forks are located.

--- Chunk 3 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 41, Chunk software\_engineering.pdf:41:4]

CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 13 uote: “More than a discipline or a body of knowledge, engineering is a verb, an action word, a way of approaching a problem.” Scott Whitmir How do we define software engineering? ? 9 For numerous additional definitions of software engineering, see www.answers.com/topic/ software-engineering#wp-\_note-13. 10 Quality management and related approaches are discussed in Chapter 14 and throughout Part 3 of this book. Software engineering encompasses a process, methods for managing and engineering software, and tools. pre75977\_ch01.qxd 11/27/08 3:11 PM Page 13

--- Chunk 4 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 125, Chunk software\_engineering.pdf:125:4]

1 Some writers argue for one of these terms to the exclusion of the others. In reality, software engineering is all three. pre75977\_ch04.qxd 11/27/08 3:27 PM Page 97

--- Chunk 5 ---

[Source: C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf, Page 613, Chunk software\_engineering.pdf:613:0]

SCM activities are developed to (1) identify change, (2) control change, (3) ensure that change is being properly implemented, and (4) report changes to others who may have an interest. It is important to make a clear distinction between software support and software configuration management. Support is a set of software engineering activities that occur after software has been delivered to the customer and put into operation. Software configuration management is a set of tracking and control activities that are initiated when a software engineering project begins and terminates only when the software is taken out of operation. A primary goal of software engineering is to improve the ease with which changes can be accommodated and reduce the amount of effort expended when changes

3)

Model: meta-llama/Llama-3.3-70B-Instruct-Turbo-Free

Prompt : prompt 1

Question : question 1

Embedding model: model\_name="Qwen/Qwen3-Embedding-0.6B",

Chunking method: semantic chunking

4)

LLM: gemini-1.5-flash

Prompt: prompt1

Embedding model: Qwen/Qwen3-Embedding-0.6B

Chunking method: semantic chunking with Qwen/Qwen3-Embedding-0.6B

Reanking :none  
📘 Answer:

Based on the provided text, software engineering is defined in several ways:

1. \*\*The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.\*\* This definition emphasizes a structured and measurable process.

2. \*\*The study of approaches as in (1).\*\* This highlights the academic and research aspects of the field.

3. Software engineering encompasses \*\*process, methods, and tools\*\* that enable complex computer-based systems to be built in a timely manner with quality. This emphasizes the practical aspects, including the software development process, techniques used, and supporting tools.

The text also notes that software engineering rests on an organizational commitment to quality and is a layered technology, with the process layer forming its foundation. It's described as a problem-solving activity following core principles and requiring adaptability and agility.

📚 Context Chunks Used:

--- Chunk 1 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 32, Chunk software\_engineering.pdf:32:1]

Software is engineered, not manufactured. pre75977\_ch01.qxd 11/27/08 3:11 PM Page 4

--- Chunk 2 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 32, Chunk software\_engineering.pdf:32:0]

programmer are the same questions that are asked when modern computer-based systems are built:1 4 CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 1 In an excellent book of essays on the software business, Tom DeMarco [DeM95] argues the coun- terpoint. He states: “Instead of asking why software costs so much, we need to begin asking ‘What have we done to make it possible for today’s software to cost so little?’ The answer to that ques- tion will help us continue the extraordinary level of achievement that has always distinguished the software industry.” • Why does it take so long to get software finished? • Why are development costs so high? • Why can’t we find all errors before we give the software to our customers? • Why do we spend so much time and effort maintaining existing programs? • Why do we continue to have difficulty in measuring progress as software is being developed and maintained? These, and many other questions, are a manifestation of the concern about software and the manner in which it is developed—a concern that has lead to the adoption of software engineering practice. 1.1.1 Defining Software Today, most professionals and many members of the public at large feel that they understand software. But do they? A textbook description of software might take the following form: Software is: (1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to ad- equately manipulate information, and (3) descriptive information in both hard copy and virtual forms that describes the operation and use of the programs. There is no question that other more complete definitions could be offered. But a more formal definition probably won’t measurably improve your under- standing. To accomplish that, it’s important to examine the characteristics of soft- ware that make it different from other things that human beings build. Software is a logical rather than a physical system element. Therefore, software has characteris- tics that are considerably different than those of hardware: 1. Software is developed or engineered; it is not manufactured in the classical sense. Although some similarities exist between software development and hard- ware manufacturing, the two activities are fundamentally different. In both activities, high quality is achieved through good design, but the manufactur- ing phase for hardware can introduce quality problems that are nonexistent How should we define software? ?

--- Chunk 3 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 42, Chunk software\_engineering.pdf:42:0]

that must be established for effective delivery of software engineering technology. The software process forms the basis for management control of software projects and establishes the context in which technical methods are applied, work products (models, documents, data, reports, forms, etc.) are produced, milestones are estab- lished, quality is ensured, and change is properly managed. Software engineering methods provide the technical how-to’s for building soft- ware.

--- Chunk 4 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 41, Chunk software\_engineering.pdf:41:1]

What are the “sound engineering principles” that can be applied to computer software development? How do we “economically” build software so that it is “reliable”? What is required to create computer programs that work “efficiently” on not one but many different “real machines”? These are the questions that continue to challenge software engineers. The IEEE [IEE93a] has developed a more comprehensive definition when it states: Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1). And yet, a “systematic, disciplined, and quantifiable” approach applied by one software team may be burdensome to another. We need discipline, but we also need adaptability and agility. Software engineering is a layered technology. Referring to Figure 1.3, any engineer- ing approach (including software engineering) must rest on an organizational com- mitment to quality. Total quality management, Six Sigma, and similar philosophies 10 foster a continuous process improvement culture, and it is this culture that ultimately leads to the development of increasingly more effective approaches to software engi- neering. The bedrock that supports software engineering is a quality focus. The foundation for software engineering is the process layer. The software engi- neering process is the glue that holds the technology layers together and enables rational and timely development of computer software. Process defines a framework CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 13 uote: “More than a discipline or a body of knowledge, engineering is a verb, an action word, a way of approaching a problem.” Scott Whitmir How do we define software engineering? ? 9 For numerous additional definitions of software engineering, see www.answers.com/topic/ software-engineering#wp-\_note-13. 10 Quality management and related approaches are discussed in Chapter 14 and throughout Part 3 of this book. Software engineering encompasses a process, methods for managing and engineering software, and tools.

--- Chunk 5 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 53, Chunk software\_engineering.pdf:53:0]

CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 25 With the exception of a passing reference, software was hardly mentioned as part of the conversation. And yet, software will make or break the SafeHome product line. The engineering effort will succeed only if SafeHomesoftware succeeds. The market will accept the product only if the software embedded within it properly meets the customer’s (as yet unstated) needs. We’ll follow the progression of SafeHome software engineering in many of the chapters that follow. 1.8 S UMMARY Software is the key element in the evolution of computer-based systems and products and one of the most important technologies on the world stage. Over the past 50 years, software has evolved from a specialized problem solving and infor- mation analysis tool to an industry in itself. Yet we still have trouble developing high- quality software on time and within budget. Software—programs, data, and descriptive information—addresses a wide array of technology and application areas. Legacy software continues to present special challenges to those who must maintain it. Web-based systems and applications have evolved from simple collections of in- formation content to sophisticated systems that present complex functionality and multimedia content. Although these WebApps have unique features and require- ments, they are software nonetheless. Software engineering encompasses process, methods, and tools that enable complex computer-based systems to be built in a timely manner with quality. The software process incorporates five framework activities—communication, planning, modeling, construction, and deployment—that are applicable to all software proj- ects. Software engineering practice is a problem solving activity that follows a set of core principles.

5)

LLM: gemini-1.5-flash

Prompt: prompt1

Embedding model: Qwen/Qwen3-Embedding-0.6B

Chunking method: semantic chunking with Qwen/Qwen3-Embedding-0.6B

Reanking : model used cross-encoder/ms-marco-MiniLM-L-6-v2

response = chain.run(inputs)

📘 Answer:

Based on the provided text, software engineering is defined in several ways:

1. \*\*The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.\*\* This definition emphasizes a structured and measurable process.

2. \*\*The study of approaches as in (1).\*\* This highlights the academic and research aspects of the field.

3. A layered technology resting on an organizational commitment to quality, with the software engineering process forming the foundation. This emphasizes the importance of a quality-focused culture and a well-defined process.

4. Software engineering encompasses a process, methods for managing and engineering software, and tools. This broader definition includes the practical aspects of software development.

In essence, software engineering is a multi-faceted field that combines technical skills, management practices, and a focus on quality to build and maintain computer software effectively. It addresses challenges like high costs, long development times, and difficulty in error detection and progress measurement.

📚 Context Chunks Used:

--- Chunk 1 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 32, Chunk software\_engineering.pdf:32:0]

programmer are the same questions that are asked when modern computer-based systems are built:1 4 CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 1 In an excellent book of essays on the software business, Tom DeMarco [DeM95] argues the coun- terpoint. He states: “Instead of asking why software costs so much, we need to begin asking ‘What have we done to make it possible for today’s software to cost so little?’ The answer to that ques- tion will help us continue the extraordinary level of achievement that has always distinguished the software industry.” • Why does it take so long to get software finished? • Why are development costs so high? • Why can’t we find all errors before we give the software to our customers? • Why do we spend so much time and effort maintaining existing programs? • Why do we continue to have difficulty in measuring progress as software is being developed and maintained? These, and many other questions, are a manifestation of the concern about software and the manner in which it is developed—a concern that has lead to the adoption of software engineering practice. 1.1.1 Defining Software Today, most professionals and many members of the public at large feel that they understand software. But do they? A textbook description of software might take the following form: Software is: (1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to ad- equately manipulate information, and (3) descriptive information in both hard copy and virtual forms that describes the operation and use of the programs. There is no question that other more complete definitions could be offered. But a more formal definition probably won’t measurably improve your under- standing. To accomplish that, it’s important to examine the characteristics of soft- ware that make it different from other things that human beings build. Software is a logical rather than a physical system element. Therefore, software has characteris- tics that are considerably different than those of hardware: 1. Software is developed or engineered; it is not manufactured in the classical sense. Although some similarities exist between software development and hard- ware manufacturing, the two activities are fundamentally different. In both activities, high quality is achieved through good design, but the manufactur- ing phase for hardware can introduce quality problems that are nonexistent How should we define software? ?

--- Chunk 2 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 42, Chunk software\_engineering.pdf:42:0]

that must be established for effective delivery of software engineering technology. The software process forms the basis for management control of software projects and establishes the context in which technical methods are applied, work products (models, documents, data, reports, forms, etc.) are produced, milestones are estab- lished, quality is ensured, and change is properly managed. Software engineering methods provide the technical how-to’s for building soft- ware.

--- Chunk 3 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 41, Chunk software\_engineering.pdf:41:1]

What are the “sound engineering principles” that can be applied to computer software development? How do we “economically” build software so that it is “reliable”? What is required to create computer programs that work “efficiently” on not one but many different “real machines”? These are the questions that continue to challenge software engineers. The IEEE [IEE93a] has developed a more comprehensive definition when it states: Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1). And yet, a “systematic, disciplined, and quantifiable” approach applied by one software team may be burdensome to another. We need discipline, but we also need adaptability and agility. Software engineering is a layered technology. Referring to Figure 1.3, any engineer- ing approach (including software engineering) must rest on an organizational com- mitment to quality. Total quality management, Six Sigma, and similar philosophies 10 foster a continuous process improvement culture, and it is this culture that ultimately leads to the development of increasingly more effective approaches to software engi- neering. The bedrock that supports software engineering is a quality focus. The foundation for software engineering is the process layer. The software engi- neering process is the glue that holds the technology layers together and enables rational and timely development of computer software. Process defines a framework CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 13 uote: “More than a discipline or a body of knowledge, engineering is a verb, an action word, a way of approaching a problem.” Scott Whitmir How do we define software engineering? ? 9 For numerous additional definitions of software engineering, see www.answers.com/topic/ software-engineering#wp-\_note-13. 10 Quality management and related approaches are discussed in Chapter 14 and throughout Part 3 of this book. Software engineering encompasses a process, methods for managing and engineering software, and tools.

--- Chunk 4 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 53, Chunk software\_engineering.pdf:53:0]

CHAPTER 1 SOFTWARE AND SOFTWARE ENGINEERING 25 With the exception of a passing reference, software was hardly mentioned as part of the conversation. And yet, software will make or break the SafeHome product line. The engineering effort will succeed only if SafeHomesoftware succeeds. The market will accept the product only if the software embedded within it properly meets the customer’s (as yet unstated) needs. We’ll follow the progression of SafeHome software engineering in many of the chapters that follow. 1.8 S UMMARY Software is the key element in the evolution of computer-based systems and products and one of the most important technologies on the world stage. Over the past 50 years, software has evolved from a specialized problem solving and infor- mation analysis tool to an industry in itself. Yet we still have trouble developing high- quality software on time and within budget. Software—programs, data, and descriptive information—addresses a wide array of technology and application areas. Legacy software continues to present special challenges to those who must maintain it. Web-based systems and applications have evolved from simple collections of in- formation content to sophisticated systems that present complex functionality and multimedia content. Although these WebApps have unique features and require- ments, they are software nonetheless. Software engineering encompasses process, methods, and tools that enable complex computer-based systems to be built in a timely manner with quality. The software process incorporates five framework activities—communication, planning, modeling, construction, and deployment—that are applicable to all software proj- ects. Software engineering practice is a problem solving activity that follows a set of core principles.

--- Chunk 5 ---

[Source: [C:\Users\dhili\Desktop\SRIP\week3\data\software\_engineering.pdf](file:///C:\Users\dhili\Desktop\SRIP\week3\data\software_engineering.pdf), Page 32, Chunk software\_engineering.pdf:32:1]

Software is engineered, not manufactured. pre75977\_ch01.qxd 11/27/08 3:11 PM Page 4