# Question Bank (*including MID-TERM questions*)

**Apart from the challenges of heterogeneity, business and social change and trust and security, identify other problems and challenges that software engineering is likely to face in the 21st century (hint: think about the environment).**

Problems and challenges for software engineering There are many possible challenges that could be identified. These include:

1. Developing systems that are energy-efficient. This makes them more usable on low power mobile devices and helps reduce the overall carbon footprint of IT equipment.

2. Developing validation techniques for simulation systems (which will be essential in predicting the extent and planning for climate change).

3. Developing systems for multicultural use

4. Developing systems that can be adapted quickly to new business needs

5. Designing systems for outsourced development

6. Developing systems that are resistant to attack

7. Developing systems that can be adapted and configured by end-users 8. Finding ways of testing, validating and maintaining end-user developed systems

**Based on your own knowledge of some of the application types discussed in section 1.1.2, explain, with examples, why different application types require specialized software engineering techniques to support their design and development.**

Different application types require the use of different development techniques for a number of reasons:

1. Costs and frequency of change. Some systems (such as embedded systems in consumer devices) are extremely expensive to change; others, must change frequently in response to changing requirements (e.g. business systems). Systems which are very expensive to change need extensive upfront analysis to ensure that the requirements are consistent and extensive validation to ensure that the system meets its specification. This is not costeffective for systems that change very rapidly.

2. The most important 'non-functional' requirements. Different systems have different priorities for non-functional requirements. For example, a real-time control system in an aircraft has safety as its principal priority; an interactive game has responsiveness and usability as its priority. The techniques used to achieve safety are not required for interactive gaming; the extensive UI design required for games is not needed in safety-critical control systems.

3. The software lifetime and delivery schedule. Some software systems have a relatively short lifetime (many web-based systems), others have a lifetime of tens of years (large command and control systems). Some systems have to be delivered quickly if they are to be useful. The techniques used to develop short-

lifetime, rapid delivery systems (e.g. use of scripting languages, prototyping, etc.) are inappropriate for long-lifetime systems which require techniques that allow for long-term support such as design modelling.

**What are the four important attributes that all professional software should have? Suggest four other attributes that may sometimes be significant.**

Four important attributes are maintainability, dependability, performance and usability. Other attributes that may be significant could be reusability (can it be reused in other applications), distributability (can it be distributed over a network of processors), portability (can it operate on multiple platforms e.g laptop and mobile platforms) and inter-operability (can it work with a wide range of other software systems).

*Decompositions of the 4 key attributes e.g. dependability decomposes to security, safety, availability, etc. is also a valid answer to this question.*

**Software engineering techniques and processes are not appropriate for all development projects. (a) List at least four characteristics of a project that would argue FOR using the software engineering approach.**

Grading: 1 point per characteristic Example characteristics: Size of project (big), number of people involved (>1), external requirements imposed, family of products required, many changes over time requested or likely to be requested, costly, large/major consequences as a result of the development. **(b) For each of the characteristics you list in part (a), describe at least one way software engineering attempts to address the concerns raised by that characteristic.**

Grading: one point per characteristic Example answers: Size [applying software process to the development; breaking the problem into parts using modularity; …], Number of people [process], external requirements [requirements document, acceptance test plan, reviews], family of products [software architecture], changes [design by information hiding, software architectures], costly [process model, reviews, reuse], consequences [quality assurance techniques

**Grading guide: 1 point per part of the question.**

**(a) Why should all the details of an application's user interface be included in a requirements specification?**
Because such details are externally visible, the user most likely cares about them, and such details DO describe what is required of the software.

**(b) What makes it seem inappropriate for including those details?**
Such details emerge from a design process, namely the user interface design process. Thus design would have to go on during requirements.

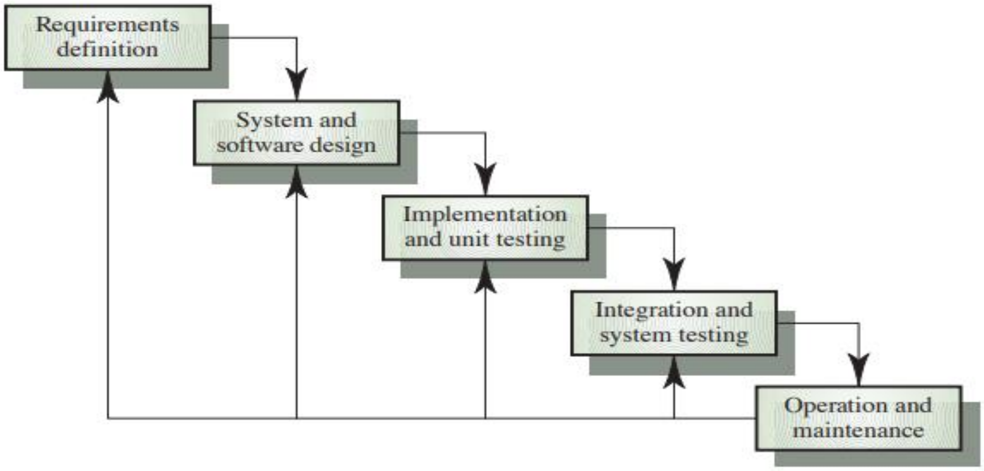
**(c) Which software process model best supports development of applications with successful user interfaces, and why?**
Several do. The Spiral model and any incremental model do. They work because the process allows users to state their initial goals and requirements, all the developers to work ahead a bit (do an initial UI design, e.g.), and then give the users the opportunity to review and rethink what they want.

**What are the three most important characteristics of a requirements specification? Explain why each of them is so important.**

Grading guide: 1 point per way. Completeness: if it is incomplete then additional requirements will be identified downstream, possibly upsetting the development entirely. Consistency: if the document is inconsistent, then what should be built? Unambiguous: if there is ambiguity a system might be built that does not satisfy the real requirements. Other possible answers: Verifiable No implementation bias.

**(a) Describe the Waterfall model of software development. Use a diagram to illustrate your version of the waterfall model.**

(Variations on the following are possible)


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

**(b) Briefly describe at least four GOOD things about the Waterfall process model.**

• Provides clear phasing of activities. • Separates requirements from design/development • Uses intermediate documents for communicating between the phases. • Allows specialists to work in phases (as opposed to someone who has to work across all phases) • It is simple • There is provision for cycling back to previous phases • (Other answers possible as well)

**(d) Describe at least two ways in which the waterfall model is a POOR guide to software development.**

• Requires all requirements to be complete before beginning design • None of the benefits of the spiral model.

**What's the purpose of the acceptance test plan?**

Provides a straightforward, unambiguous way for the customer to ascertain whether the delivered system meets the requirements. It can also be used by the developers to help assess whether they have met the requirements.

**Your company, Crazy Software Ventures (CSV for short), has decided to develop a new word processor called BeyondWords (BW for short). In its desktop version BW will compete directly with Microsoft Word and similar products from other companies. BW is also intended for use on engineering workstations (running Solaris or Linux), and versions of BW will be required for PalmPilots and Compaq IPAQ's. Some of the people in CSV's marketing division think that a version of BW should even be available for cell phones. Whether or not this is an astute business decision, you have been named project manager. Your responsibility is to establish the detailed process by which BW will be created and lead the development team. Naturally your management wants BW on the market as soon as possible and they have a limited development budget. Grading guide: 4 points per section (a, b, c)**

**(a) Will you have an extensive requirements phase? Why or why not? What will you emphasize in the requirements phase?**

You could successfully argue either side of this. Here's a couple of possible arguments.

(i) For extensive requirements. This is clearly going to be a tough market to enter: you're competing against MS in the desktop market, and trying to put functionality on some very limited platforms in other markets. It stands to reason, therefore, that you should very carefully consider the ways in which you might succeed. A careful requirements analysis might reveal specific features or capabilities that you could provide in these markets that would generate sales.. presumably features not found in MS Word. Or even if they are found in Word, if you can implement them in a small fraction of the space required by Word, you might win.

(ii) Against extensive requirements. Everyone knows what features a Word processor must have on desktop platforms, so why spend a lot of time documenting them? On the other hand, implementing such features on small platforms could prove a considerable challenge, especially with regard to their user interface/usability features. In this case, moving quickly into some exploratory architecture and generating some little prototypes might help reveal the key issues you have to focus on. So… you could call this requirements analysis in the sense of the Spiral model. Note that in either case, your emphasis has to be on learning. Learning what features you must have, what features you can compete in the marketplace with, learning what are the killer issues in implementing BW on the various platforms.

**(b) What qualities will you emphasize in the product and why?**

My first choice would be usability. CSV is clearly targeting a wide range of users, but certainly "ordinary consumers" are part of the list. How will you win against MS except by having better (or at least different ) functionality that is easier to use? I would tend to downplay quality, in the sense of correctness, initially. This somewhat goes against the grain, but this is a high-risk venture (that's why the company is known as CSV), and if you're going to succeed it is going to be because of good buzz --- great features that are easy to use. Besides, customers are used to MS failing at unpredictable times in inappropriate ways, so you could get away with some of the same low-quality. (Alternatively you could play the "this is rock solid, anytime, anywhere" game. But you're likely to run into problems with your management --- who want the product out the door in a hurry, or technically, since this will be hard to achieve, especially on a platform like the iPAQ.) Part of this is also likely to be speed. If you have a slow running editor then no-one is going to buy it

**Giving reasons for your answer based on the type of system being developed, suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems:**

• A university accounting system that replaces an existing system

• An interactive travel planning system that helps users plan journeys with the lowest environmental impact

University accounting system This is a system whose requirements are fairly well-known and which will be used in an environment in conjunction with lots of other systems such as a research grant management system. Therefore, **a reuse-based approach** is likely to be appropriate for this.

Interactive travel planning system System with a complex user interface but which must be stable and reliable. **An incremental development approach** is the most appropriate as the system requirements will change as real user experience with the system is gained.

**Suggest why it is important to make a distinction between developing the user requirements and developing system requirements in the requirements engineering process.**

There is a fundamental difference between the user and the system requirements that mean they should be considered separately.

1. The user requirements are intended to describe the system's functions and features from a user perspective and it is essential that users understand these requirements. They should be expressed in natural language and may not be expressed in great detail, to allow some implementation flexibility. The people involved in the process must be able to understand the user's environment and application domain.

2. The system requirements are much more detailed than the user requirements and are intended to be a precise specification of the system that may be part of a system contract. They may also be used in situations where development is outsourced and the development team need a complete specification of what should be developed. The system requirements are developed after user requirements have been established.

**Discover ambiguities or omissions in the following statement of requirements for part of a ticket-issuing system: An automated ticket-issuing system sells rail tickets. Users select their destination and input a credit card and a personal identification number. The rail ticket is issued and their credit card account charged. When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select a destination. Once a destination has been selected, users are requested to input their credit card. Its validity is checked and the user is then requested to input a personal identifier. When the credit transaction has been validated, the ticket is issued.**

Ambiguities and omissions include:

1. Can a customer buy several tickets for the same destination together or must they be bought one at a time?

2. Can customers cancel a request if a mistake has been made?

3. How should the system respond if an invalid card is input?

4. What happens if customers try to put their card in before selecting a destination (as they would in ATM machines)?

5. Must the user press the start button again if they wish to buy another ticket to a different destination?

6. Should the system only sell tickets between the station where the machine is situated and direct connections or should it include all possible destinations?

**Using your knowledge of how an ATM is used, develop a use case that could serve as a basis for understanding the requirements for an ATM system.**

**Withdraw cash:**

Actors: Customer, ATM, Accounting system Inputs: Customer's card, PIN, Bank Account details

Outputs: Customer's card, Receipt, Bank account details

Normal operation: The customer inputs his/her card into the machine. He/she s promoted for a PIN which is entered on the keypad. If correct, he/she is presented with a menu of options. The Withdraw cash option is selected. The customer is promoted with a request for the amount of cash required and inputs the amount. If there are sufficient funds in his/her account, the cash is dispensed, a receipt if printed and the account balance is updated. Before the cash is dispensed, the card is returned to the customer who is prompted by the machine to take their card.

Exception: Invalid card. Card is retained by machine; Customer advised to seek advice. Incorrect PIN. Customer is request to rekey PIN. If incorrect after 3 attempts, card is retained by machine and customer advised to seek advice. Insufficient balance Transaction terminated. Card returned to customer.

**MCQ:**

In Iterative development approach, which phase takes the least cost?

[A]Iterative development

[B]System Testing

[C]Specification

[D]Design

What are generic software process models? (Choose one)

[A]Waterfall, Iterative development, Agile development

[B]Waterfall, Component-based software engineering, Agile development

[C]Waterfall, Iterative development, Extreme programming development

[D]Waterfall, Iterative development, Component-based software engineering

What is a software process model?

[A]A simplified representation in a waterfall

[B]A simplified representation in a cycle of a software process

[C]A simplified representation in a step by step of a software process

[D]A simplified representation of a software process, presented from a specific perspective

The current trends suggest that the economies of ALL developed and developing nations are dependent on software

[A]False

[B]True

In Waterfall approach, which phase takes the most cost?

[A]Integration and testing

[B]Design

[C]Specification

[D]Development

What is a software process?

[A]A set of cycles whose goal is the development or evolution of software

[B]A set of steps whose goal is the development or evolution of software

[C]A set of models whose goal is the development or evolution of software

[D]A set of activities whose goal is the development or evolution of software

In Waterfall approach, which phase takes the least cost?

[A]Development

[B]Integration and testing

[C]Design

[D]Specification

In Component-based software engineering approach, which phase takes the least cost?
[A]Specification

[B]Design

[C]Development

[D]Integration and testing

What are the key challenges facing software engineering in the 21st century? (Choose one)

[A]The heterogeneity challenge, the trust challenge, the life time challenge

[B]The heterogeneity challenge, the delivery challenge, the trust challenge

[C]The delivery challenge, the trust challenge, the life time challenge

[D]The heterogeneity challenge, the delivery challenge, the life time challenge

Which is the correct sequence of the following activities in the process of System evolution?

(1) Assess existing systems (2) Define system requirements (3) Modify systems (4) Propose system changes

[A]2=>1=>3=>4

[B]2=>1=>4=>3

[C]2=>3=>1=>4

[D]2=>3=>4=>1

What is the main difference between the Spiral model and other software processes models?

[A]The explicit recognition of risk in the Spiral model

[B]The goal of Objective setting phase

[C]The goal of Development and validation phase

[D]The goal of Planning phase

What is the outputs of Requirements elicitation and analysis? (Choose one)

[A]System models

[B]User and system requirements document

[C]Requirements document

What is the output of Feasibility study phase in the requirements engineering process?
[A]User requirement document

[B]Report that recommends whether or not to continue development project

[C]Stakeholders list

[D]Project plan

Which is the correct sequence of the following activities in the Waterfall model?

(1) System and software design (2) Requirement definition (3) Integration and system testing (4) Implementation and unit testing (5) Operation and maintenance

[A]1=>4=>3=>2=>5

[B]1=>2=>3=>4=>5

[C]2=>1=>4=>3=>5

[D]2=>3=>4=>1=>5

Which is the correct sequence of the following activities in the Component-based software engineering?

(1) Component analysis (2) System design with reuse (3) Requirements modification (4) Development and integration

[A]1=>3=>2=>4

[B]1=>3=>4=>2

[C]1=>2=>4=>3

[D]1=>2=>3=>4

The Waterfall approach is the BEST approach to development software when __

[A]Adding some new features to a system

[B]Un-stable requirement systems

[C]The software system is small, unclear requirements

[D]The software system is large and that is need developed at several sites

What are non – functional requirements?

[A]Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc

[B]Requirements that come from the application domain of the system and that reflect characteristics and constraints of that domain

[C]Statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situations.

What are good attributes of requirements? (Choose one)

[A]Testable, complete, clear, inconsistent, unambiguous

[B]Testable, complete, clear, consistent, ambiguous

[C]Testable, incomplete, clear, consistent, unambiguous

[D]Testable, complete, clear, consistent, unambiguous

What are domain requirements?

[A]Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc

[B]Statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situations.

[C]Requirements that come from the application domain of the system and that reflect characteristics and constraints of that domain

What are system requirements?

[A]A structured document setting out detailed descriptions of the system‟s functions, services and operational constraints

[B]Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc

[C]Statements in natural lang

Which are types of non-functional requirement? (Choose one)

[A]Product requirements; Organizational requirements; External requirements

[B]Organizational requirements; External requirements; Non-Functional Requirements

[C]Organizational requirements; External requirements; System Requirements [D]User requirements; Organizational requirements; External requirements

What is the BEST way to write requirement document?

[A]Should include HOW the system will be implemented in a specification of the system requirements

[B]Should NOT include a definition of user requirements in a specification of the system requirements

[C]Requirement document should set of WHAT the system should do and HOW it should do

[D]Should include both a definition of user requirements and a specification of the system requirements

What are functional requirements?

[A]Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc

[B]Requirements that come from the application domain of the system and that reflect characteristics of that domain

[C]Statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situations.

What are user requirements?

[A]Statements in natural language plus diagrams of the services the system provides and its operational constraints

[B]Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc

[C]A structured document setting out detailed descriptions of the system‟s functions, services and operational constraints

Which of the following requirement statements belongs to non-functional requirements?

[A]An article information shall include Author, Title, Description and other related information

[B]The user interface shall be implemented as simple HTML without frames

[C]The system shall be able to search either all of the initial set of databases or select a sub set from it,

[D]The system shall provide appropriate viewers for the user to read documents in the document store