Pergunta 1

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

IEEE defines software engineering as "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software".

Which concerns are in your opinion the most important in software engineering? Why?

a) coding, testing, and deploying software

b) managing stakeholders and the team

c) leading and managing the process from the initial idea to the final software system

d) defining and enforcing good values, principles and practices capable to produce good software

Option d
If good values, principles and practices are being applied, the end product is sure to be good. These points allow for better communication, understanding and better team work. No matter what developing technique, team or product being developed, if the developers have been taught the same practices, they are sure to understand each other's work better, work well together and come up with a better end product.

Pergunta 2

Respondida Pontuou 3,000 de 5,000 Destacar pergunta

RUP is not a prescriptive software development methodology, but instead a framework. What do you think was the motivation from its authors? Elaborate what can vary between instances of RUP.

a) Since traditional development was slow at starting to deliver software, being a framework, RUP delivers software incrementally and iteratively, resulting in earlier value for the client.

b) It is typically very hard to follow strict rules. As a framework, the team is still able to decide how they want to collaborate.

c) Prescriptive methodologies require a large amount of documentation. As a framework, the amount of documentation required can be adjusted to accelerate the beginning of the implementation phase.

d) Each software development project has its own intricacies. Being a framework enables rUP to be adapted to several projects without imposing strict rules that would become inefficient for some contexts.

Option d
(assumign that by instances of RUP we are reffering to different projects RUP can be applied to:)
RUP allows for compromise between the developing team and the client, since the client can state what he wants and is paying for, but the team has the space to discuss how it should be done and is realistic to do at each iteration. The fact that there is a framework allows for both parties to get a sense of what has to be discussed and decided.

Pergunta 3

Respondida Pontuou 0,000 de 5,000 Destacar pergunta

eXtreme Programming is praised by its software development efficiency. Pair programming is one of the practices contributing to said efficiency. What does pair programming consists of? Briefly detail the practice using your own words.

a) Two developers working on their own features, but assigned to help each other;

b) One developer working on part of a feature, while the other tackles the other part, merging their implementation afterwards;

c) One developer working on a feature, while being continuously reviewed by another, often trading their roles;

d) Two developers independently working on the same feature, discussing the two implementations and choosing the best solution.

Pergunta 3

Respondida Pontuou 0,000 de 5,000 Destacar pergunta

eXtreme Programming is praised by its software development efficiency. Pair programming is one of the practices contributing to said efficiency. What does pair programming consists of? Briefly detail the practice using your own words.

a) Two developers working on their own features, but assigned to help each other;

b) One developer working on part of a feature, while the other tackles the other part, merging their implementation afterwards;

c) One developer working on a feature, while being continuously reviewed by another, often trading their roles;

d) Two developers independently working on the same feature, discussing the two implementations and choosing the best solution.

Option d
Pair programming consists of two programmers working on the same feature, but comming up with different solutions to solve that feature. So you have two different implementations that were implemented by two different people, but then they are discussed between the pair to come up with the final best solution. In the end, that code still belongs to the whole team and not the single developer who first came up with it.

Pergunta 4

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Meeting project deadlines is possibly the most recurring failure of software development. How does Scrum tries to prevent this? Justify.

a) Agile methodologies will only ensure a product increment each sprint. Traditional methodologies are more strict, hence, more capable of meeting deadlines;

b) The ScrumMaster is responsible for ensuring the team is efficient and delivers the software at the right time;

c) Effort estimations and previous sprint velocity can be used to extrapolate when the backlog will be complete;

d) A well defined process will ensure development efficiency, which will guarantee the project deadline.

may have access to the increments throughout the development process. That in itself, even if the team doesn't meet the deadline, helps the customer have an idea of the progress being made, instead of just arriving at the supposed delivery day and not have a product to show. Also, in Scrum, there are estimations of how long a sprint takes and each sprint should add to a working increment of the project, so it is easier to estimate and predict how long it will take for each delivery. This way, it is easier to stay away from agreeing to irrealistic deadlines and, therefore, the team is more likely to deliver the product on time.

Pergunta 5

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Requirements documents have different target audiences, from customers to engineers.

What should a good software requirements document include?

a) the detailed conditions of the contract between the customers and the engineers

b) the what and how the system should do

c) the functional and nonfunctional requirements in detail

d) the system models showing the relationships between the components and its environment, and a solution satisfying them.

Option c

The functional and nonfunctional requirements are the most fundamental components to be included in a software requirements document, since those are the actual specification/characteristics for the product being developed. Although system models are quite useful and can be included too, the functional and non functional requirements are what's indespensable since they allow both customer and developing team to be sure of what is being agreed to, concerning what the system should do.

Pergunta 6

Respondida Pontuou 0,000 de 5,000 Destacar pergunta

Software Engineering has been a research subject for the past decades. Focusing on software architectures, engineers have learned much about how to build good software. What technique/approach is typically adopted to share such knowledge between engineers? Describe why is it a good technique/approach.

a) Object Oriented Programming.

b) Reusable Open Source libraries.

c) Design Patterns.

d) Sequence Diagrams.

Option b
Reusable Open Source libraries are undoubtedly the best approach to share knowledge between engineers, since not only we are providing each other with working copies of code that can be studied, but also giving access to a valueable tool for faster implementation. If I do not have to spend time implementing what has already been implemented, I have more time to implement new things and, later on, I may share what I have implemented too and this becomes a cycle. So, not only there is knowledge being shared between engineers, but also time being saved that can be invested into

Pergunta 7

Respondida Pontuou 0,000 de 5,000 Destacar pergunta

Legacy systems are older software systems, developed using obsolete software and hardware technologies, that remain useful for a business. It is often cheaper and less risky to maintain a legacy system than to develop a replacement system using modern technology. What do you think is the main reason for this?

a) Modern technology is unable to cope with old requirements.

b) Hiring a new team, proficient in the modern technologies, is too expensive.

c) The business value of the legacy system still outweighs the technology value.

d) The maintenance team has too much implicit knowledge on the system.

Option b

I believe companies decide to keep their legacy systems because hiring a team to develop new software from start is usually more expensive than having maintenance done to the system.
Also, it might be risky, since new technology is always being developed and systems quickly become outdated again. In addition, what if the previous system worked better? And how about having to teach employees to work with a different system?
Ultimately, the business value and quality of the technology should be taken into consideration.

Pergunta 8

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Test-driven development promotes smaller, terse and more balanced codebases. What activities/benefits do you think contribute the most to this outcome?

a) "Test-first" and refactoring.

b) Regression testing and automated tests.

c) Code coverage and system documentation.

d) Automated Tests as acceptance criteria and regression testing.

Option a

I believe both automated tests and refactoring are what contribute the most to smaller, terse and balanced codebases. Automated tests as acceptance criteria save a lot of time by allowing the developer to know whenever a change to the program, no matter how small. changes ripple effects that may interfere with the correct functioning of said program. Refactoring allows for smaller codebases where code can be reused instead of having several different pieces of code that do the same thing and could be resumed into one single function.

Pergunta 9

Respondida Pontuou 3,000 de 5,000 Destacar pergunta

Software project management is concerned with activities to ensure that software is delivered on time, on budget, and in accordance with the requirements of the organisations developing and procuring the software. One of the activities of project management is planning the software releases: which features to deliver, and when.

Which of the following do you consider the best practice to plan a software project?

a) the team elements self-organize to select the features to include in the next release

b) the developers estimate the time to implement a set of features

c) the project manager defines the features and the effort required to implement the releases

d) the team defines what to implement in the next release and the project defines how to implement it

Option c
I believe having a project manager defining both the features and effort required to implement the releases is the best option, since that person is in close contact with the developing team in order to know what they are capable of and is realistic for them to do, but also is distant enough of the tasks at hand that he can make a fair decision.
The project manager is the common link between the developing team and the client and, therefore, the only person qualified to plan, schedule and

Pergunta 10

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Software bugs are unfortunate but happen very frequently. Bugs must be fixed. Sometimes, bugs are fixed by the original development team. Other times, bugs are fixed by dedicated bug-fixing teams. In open source projects, the situation is more extreme, with teams made of a crowd of developers, possibly worldwide distributed, that can't meet in person, or talk, and have very short, or even none documentation, to exchange knowledge.

With regards to your experience of bug fixing in the context of the T34, describe how did you manage the process to do the fix, from the idea to the final code?

The T34 work was extremely complicated. To start with, the instalation process was an absolute nightmare and once everything was up and running, we found out that there were worst nightmares to deal with. Choosing a bug out of thousands, having those bugs being fixed when we were halfway done solving them ourselves... And then having new bugs added systematically! Working with open source software is not an easy task!
Our thought process had to become: We'll choose the most recent thing added so we, hopeffuly, get to solve it before someone else does. Even if someone else solves it, we're going to present our solution to the teacher and that's it. We had to study the code, understand what was being asked to