Pergunta 1

Software is getting extremely pervasive in our life, and at the same time is getting harder to build with quality and efficiently. Since many years ago, the computing power had outpaced the ability of software engineers to effectively use those capabilities, resulting in many problems.

Which are in your opinion the problems that were never solved up until our days?

a) Projects running over-budget, over-time, and unmanageable

b) Software very inefficient and with low quality

c) Software often not meeting requirements

d) Software never delivered

Pergunta 2

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

The Rational Unified Process was introduced in the early 2000s. Choose the best description for RUP and shortly elaborate about it in your words.

a) An iterative software development process framework;

b) An iterative prescriptive development process;

c) A set of guidelines for using UML to capture requirements and interact with clients;

d) A framework for capturing and monitoring the implementation of software requirements.

Option a). RUP is a software development process which is divided in cycles, phases and iterations. The functionality of the system is delivered after a series of releases of increasing completeness. In each iteration, different requirements are selected to be developed, providing new releases every iteration. Therefore, we can conclude that RUT is an interative development process framework.

Pergunta 2

Respondida Pontuou 5,000 de 5,000 Retirar destaque

RUP is composed by several engineering disciplines. Analysis and design is one of those disciplines. What roles are required by this discipline? Elaborate what they use as input and what is the output of their work.

a) The whole development team;

b) Lead developer and clients;

c) Requirements engineers and product manager;

d) Software architects and designers.

d) the architects perform an architectural analysis, that is then reviewed by the designers; The architects then develop the architectural design, describe concurrency and distribution, and later review their work before sending it back to the design team; the design team then implements the subsystem and class design, as well as the use case design; All work is then reviewd.

Pergunta 3

Respondida Pontuou 5,000 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.

Answer c)
As stated above, pair programming is a software development practice, where two developers work with only one computer, trying to minimize human error, increase productivity and maximize code-base knowledge. I can't think of an equivalent idiom in english, "Duas cabeças pensam melhor que uma".

Pergunta 3

eXtreme Programming is an iterative methodology that values simplicity and adaptability to change. Despite its advantages, it is not ideal for all software projects. Choose which of the following projects you believe XP would not applicable, justifying your choice.

a) Designing a large corporation website;

b) A project with a strict deadline;

c) A research project for an heart implant;

d) A large scale micro-service cloud application.

Pergunta 4

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

The Agile manifesto defends:

```
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan
```

In your opinion, what did the authors wanted to achieve by introducing the manifesto? Justify your option.

a) A faster way to develop software, by discarding the need for documenting requirements and source code;

b) A more efficient software development, by adopting novel development architectures and tools;

c) A cheaper way to develop software, by eliminating overheads and unnecessary roles;

d) A less bureaucratic approach to develop software by bringing clients and development team to work better and together.

Option d). Agile methods focus more on people than process and tools. With a costumer viewing the delivery of increments in each iteration, he can get and provide feedback towards the product work. This creates trust between the clients and the developers, which forms a positive culture where both sides expect the project to succed. This is possible due to the less formal apporach of the agile methods.

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.

c) I feel that "c" is the best representation of why SCRUM is so effective at keeping its schedules correct. Knowing before hand how much work can be done in each sprint keeps the developement time honest and accurate.
Additionaly and very importantly, knowing how long developement time is (from analysing velocity) allows the team to alocate more or less work/manpower to each subsquent sprint to compensate for their previous delays.

Pergunta 4

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Scrum motivate a close relationship between the client and the development team. Choose the reason why, justifying your choice by elaborating on the relevance of the client's involvement.

a) To provide additional details left out in the requirements during the sprint;

b) To ensure that the team works at the expected velocity;

c) To validate the implementation and provide feedback for future sprints;

d) All of the above.

Answer c)

By doing that the client can state were he/she's please with the project direction, and gets to know, get's used to, the project from an early stage. It also builds trust between the client and the team, with the prospect that the project will become successfull.

Pergunta 5

Respondida Pontuou 0,000 de 5,000 Destacar pergunta

Requirements engineering refers to the process of defining, documenting and maintaining requirements in the software engineering process. There are different kinds of activities involved, and different types of requirements.

Other important characteristics of the process of requirements engineering are:

a) Requirements engineering comprises several activities, from elicitation to validation, being this last one the easier to perform.

b) Requirements can be categorized as functional, also known as user requirements, and non-functional, also known as system requirements, and domain, also known as application requirements. All kinds are equally important.

c) Requirements engineering is very valued in waterfall-like processes, being always the first phase of the development process. More recent processes assume the importance of requirements engineering along all the process, even at usability studies phases.

d) Requirements are ignored in almost all agile methods, considered a waste of time and effort, since there is no clear evidence that their identification contributes to successful systems.

Answer: d)

Agile methods ignore system requirents and documentation to be able to have a shorter, less strict development cycle

Pergunta 5

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Requirements elicitation and analysis include discovery, organization, priorisation, and specification of requirements. It seems a paradox, but it is common to say that stakeholders don't know what they really want, or don't know how to express it, and usually do not agree, creating requirements conflicts.

Which of the following practices do you consider to be important to do for an effective requirements elicitation?

a) Specify the requirements in the most natural way as possible, to use the same language of the stakeholders.

b) Specify the requirements in the most structured format as possible, to eliminate all conflicts.

c) Specify the requirements as use cases, or user stories, to illustrate all possible interactions with the system.

d) Specify the requirements at the beginning of the project only, to avoid later changes.

Option c). Establishing the system requirements involve technical staff working with customers or other stakeholders to find which services/functionailities should the system provide, the application domain, the work enviroment and operational constraints. Since it involves such a large variety of people it is better to specify the requirements with use cases models, user stories since stakeholders can relate to them and comment on their situation with respect to the story/case. This will avoid conflicts provided by language or technical formats not commonly used by costumers/stakeholders.

Pergunta 6

Open Source software is software that can be freely accessed, used, changed, and shared (in modified or unmodified form) by anyone. Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition.

Which of the following statement is false?

a) It is illegal to build and sell software that reuses open-source libraries;

b) Open-source software can be used for personal financial reasons;

c) A software is only open-source if distributed with an open-source license;

d) Open-source licenses can be used to while creating commercial software.

Pergunta 6

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Software is complex to build. If we were to build all our applications from scratch, software engineering would be an inefficient discipline. Fortunately, software can be composed and reused, enabling us to take on existing software as a scaffold upon which we can build our own.

When building and running the program below in C, which level of software reuse is being applied?

```
/* Hello World program */
#include<stdio.h>
main()
{
    printf("Hello World");
}
```

a) Component level;

b) Abstraction level;

c) Design level;

d) Object level.

Answer: d)
We're looking at the use of C's standard library , a programming language library.

Pergunta 7

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

What is the main cause for software degradation?

a) Maintenance.

b) Lack of Re-engineering.

c) Poor Refactoring.

d) Low Technical Debt.

Answer: c)

Bad refactoring lead unmantainable code base, code changes and added new features with no refactoiriung may lead to code smells, code duplication and bad software design

Pergunta 7

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

It is common knowledge in Software Engineering that "software must evolve, or it will die". Evolving software is thus a key practice to ensure software longevity. What do you think is a major factor of entropy to evolving a system?

a) The development process used to construct the system.

b) The change proposals.

c) The existing development documentation.

d) The evolution team.

Option c). Since projects tend to be developed using methodoligies such as agile methods which tend to minimize documentation, it gets harder with each change to understand how the software is organized and structured, making it harder to evolve.

Pergunta 8

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Acceptance testing is a user testing process where the aim is to decide if the software is good enough to be deployed and used in its operation environment. But doing acceptance testing is not easy and straightforward, mainly because:

a) There is no automation.

b) It can only to be run after the whole system is developed and installed.

c) It must cover the needs of all stakeholders.

d) It doesn't cover usability issues.

Option c). Since the user/customer work with the development team to define the tests for the is integrated with the develoipment, it is hard to know if the user can cover the needs of all the stakeholders.

Pergunta 8

Respondida Pontuou 5,000 de 5,000 Destacar pergunta

Acceptance testing is a user testing process where the aim is to decide if the software is good enough to be deployed and used in its operation environment. But doing acceptance testing is not easy and straightforward, mainly because:

a) There is no automation.

b) It can only to be run after the whole system is developed and installed.

c) It must cover the needs of all stakeholders.

d) It doesn't cover usability issues.

c) There is not garentee that the user is typical and is representing the interest of all stakeholders, therefore it is complicated to scale the issue the user has to more cases.

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.

Answer a)

By firstly writting the test were are the requirements of what we're implementing are met, If the test is already passing, you know your solution is already a valid one, this way you won't overthink or over engineer, this will make commonly the codebase smaller. It also helps out in refactoring, because you can refactor without worring about ruinning your code If the test was ok and now isn't, it means you changed some thing the wrong way, and with that in mind you can fix it swiftly.

Pergunta 9

Respondida Pontuou 5,000 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

Answer b)

By doing that de manager can more accuractly estimate the required time for the next release and plan out with the team the feasibility of said features.

Pergunta 9

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

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?

Initially, we started by understanding what was the bug and why was it happening. Then, we have settled the requirements for that functionality, in other words, what should happen instead. Due to the lack of information, we started by analysing the code focusing on re-engineering strategies. In order to make it easier to tackle the problem, we have desinged a UML diagram which described the relationship between all the required object classes for the fix. With this model, we saw how we could fix the bug and started coding. The final step was verifying that the bug was fixed and commiting the result.

Pergunta 10

Respondida Pontuou 4,000 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?

After choosing our issues we firstly tried to get used to the code base, to know which components were responsible for what in the system. Then we discussed the best aproach to solve the problem, aftewards we implemented our solution, commited and submited a pull requeset