Software Engineering and Development I

Index

Generated on 28.05.2023 20:40:25 with 135 questions

Software Engineering

Ian Sommerville defines Software Engineering as...
an engineering discipline that is concerned with all aspects of software production (from initial conception to operation and maintenance)

What are the fundamental software engineering activities?
Software specification, software design and development, software validation and software evolution

Software dependability includes a range of characteristics including reliability. What does this mean?
Dependable software should not cause physical or economic damage in the event of system failure

What are the attributes of good software?
Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable

Software should be efficient. What does this mean?
That it should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, resource utilization, etc.

For custom software, evolution costs often exceed development costs. Why?
Because requirements change and evolve all the time. And so must software, as a consequence.

Software dependability includes a range of characteristics including security, and safety. What does this mean?
Software has to be secure so that malicious users cannot access or damage the system

What differences has the Internet made to software engineering?
Not only has the Internet led to the development of massive, highly distributed, service-based systems, it has also supported the creation of an “app” industry for mobile devices

Software products may be developed for different general targets. What are they?
The general market or a particular customer

Software should be maintainable. What does this mean?
It means that is must be written in such a way that it can evolve to meet the changing needs of customers

What is the difference between software engineering and system engineering?
System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

What are the best software engineering techniques and methods?
While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system

What is the difference between software engineering and computer science?
Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software

What are the costs of software engineering?
Roughly 60% are development costs, 40% are testing costs

Compared to the more traditional engineering disciplines, we can say that Software Engineering is... (1 word)
young

What are the key challenges software engineering must face?
Increasing diversity, tighter delivery schedules, and dependable software

Ian Sommerville defines software as...
computer programs and associated documentation

Software must be acceptable to the type of users for which it is designed. What does this mean?
This means that it must be understandable, usable, and compatible with other systems that they use

Software Processes

What are the 3 phases of the improvement cycle of software processes?
Measure, analyze, modify.

What is that one constant and underlying element in all software processes, that must always be accounted for (1 word)?
Change

Software evolution is that software process that takes place... when?
When software systems must be changed to satisfy new requirements

One of the general process models is the waterfall model. Describe it briefly in one sentence.
A model that organizes the processes activities in separate and sequential phases and delivers the final system only at the end

One of the general process models is the integration and configuration model. Describe it briefly in one sentence.
A model that delivers the final system by reusing existing component, by configuring and integrating them based on the context of the project

One of the general process models is the incremental model. Describe it briefly in one sentence.
A model that organizes the processes activities in interleaved phases and develops and delivers the system in subsequent versions (increments)

Software engineers should always aim at improving software processes to improve the quality of software products, reduce development time and effort, and reduce costs. What should software processes refinement and change be based on?
Measurements, objective data and their analysis

What are the three main general process models?
Waterfall model. Incremental model. Integration and configuration model.

What do general process models define?
The approach to software development. The organization of software processes.

Iterative process models approach change with... (1 word)
tolerance

Software design and development are those processes that ... serve what purpose?
Turn software specifications into executable software systems

Why is prototyping useful in certain circumstances?

Because it can be used to check requirements and/or feasibility and it helps anticipating change

Software processes are the set of activities aimed at ... software systems. What is the missing word?
producing

What does the waterfall process model approach change with?
With (very) little flexibility

Software validation deals with...
making sure that a software system complies with requirements and that it actually sastifies users' needs

Configuration management

Change Management is the process that is concerned with...
all of the activities around proposing, evaluating and implementing changes to a software system

Configuration management (CM) is concerned with...
...the policies, processes, and tools for managing changing software systems

What is a "system release"?
It is a version of a software system that is distributed to customers

System building is the process of...
creating a complete, executable system by compiling and linking the system components, external libraries, configuration files, and other information

Version management is the process of...
...keeping track of different versions of software components and the systems in which these components are used

Software Versioning

I want to make sure I have all my team mate's latest changes in my dev branch. What is the GIT command I use?
git pull

What does the following GIT command do? git clone <URI>
It creates an identical local copy (clone) of the repository found at the given URL

GIT does not allow multiple developers to work simultaneously at the same software component? True/False? Why?
False. GIT keeps track of changes and guarantees that they are merged (automatically, if possible, or by signalling conflicts and supporting their manual resolution, otherwise)

In what way a Version Control System distinguishes different versions of the same component?
By assigning each version a unique ID

I want to send my changes upstream with GIT. What is the command I use?
git push

From an architectural point of view, a modern Version Control System belongs to one of two types. Which ones?
Centralized or distributed

What is a branching model to a software development team?
It is a set of conventions and procedures team members agree on and follow in order to come to a managed software development process

I am implementing a new feature and I want to avoid running the risk of losing my work. What do I do?
I push my feature branch onto a remote main repository so as to effectively create a shared backup and keep it synced as I make changes

Two software engineers approach the implementation of a feature with GIT in the following different ways.
Whose approach is best and why?
DEV1

```
git checkout dev
git pull
git checkout -b feature-1234
...
git add .
git commit -m "done feature 1234"
git checkout dev
git pull
git merge feature-1234
git push
DEV2
git checkout dev
git pull
...
git add .
git commit -m "done feature 1234"
git pull
git push
```

DEV1 conveniently isolates the changes in an separate branch

In what way a Version Control System minimizes the use of storage space?
By ensuring that duplicate copies of identical files are not maintained

Software Dependencies

A "Software dependency" is simply defined as...
Whatever piece of code that is relied upon for a digital service to work properly

I need a password hashing algorithm. I decide to develop my own. Am I making a good decision? Why?
No. The problem has already been solved. My implementation can only be worse than those available.

In developing a software product I want to re-use as much existing code as possible. Indicate at least two reasons why this statement is true or false.
It is true. Re-using existing code speeds up development and reduces the risks of bugs.

In a Java context, what is the name of a dependencies management tool?
Maven

Software dependencies raise an important issue. What is it?
Trust

For the software product I am developing I need a mathematical library. I decide to download one and "vendor" it in my project structure. Is this a good approach?
Rare cases excluded, this is a bad choice. The use of a dependencies management tool is in general a better choice.

A software dependencies management tool typically relies on an Internet repository. What risk does this entail and how can it be mitigated?
The risk is linked to the availability of the repository. We can therefore possibly create our own mirror.

For a given software product I am developing, I define a dependency without fixing its version number. My dependencies management tool will therefore always download the latest available version. When and why is this a good decision and when is it not?
It is a good decision only if I can test the dependency systematically (functionality, security, perforcance, ...). Otherwise it is a bad decision.

What is a "transitive dependency"?
A dependency of a dependency

What is "dependency hell"?
It is a phenomenon in which dependencies cannot be resolved

Software Build

With mvn (Maven) what is the phase normally linked to the building of the final distributable artifact?
package

What is the difference between a statically linked and a dynamically linked executable object?
A statically linked executable packages all its dependencies internally. A dynamically linked executable contains external references to its dependencies, which are then loaded at run-time as needed.

The build process typically involves 3 systems. Which ones?
Development, build and target systems

A software system must always be built from a clean environment. True/False? Why?
True. A clean environment guarantees the build process is not influenced (positively or negatively) by unwanted dependencies

It is alway advisable to build a software system locally before sending a new version upstream. True/False? Why?
True. We want to minimize the risk of sharing a non-buildable (broken) system

With mvn (Maven) what must I configure if I want to distribute my system as an executable .jar file?
A plugin, that allows to assemble my system with all its dependencies

What does a waterfall approach to building a software system minimize?
The number of components to be built, thus minimizing the resources and the time needed

It is always important to automate the build process of a software system. True/False? Why?
True. Building is a very complex process. Automation reduces the risk of manual/human errors and guarantees repeatability

In a C environment, what is an available build tool?
make

In mvn (Maven) what section of the configuration file refers to the building of the artifacts?
The <build> section

Requirements engineering

Look at the two requirements defined below by engineers ENG1 and ENG2. Whose definition is best and why?
ENG1
The password reset procedure must be fast and user-friendly.

ENG2
Users must be able to reset their passwords autonomously.
The procedure must be guided and take at most 2 minutes to complete.
ENG2. She defines a clear and measurable requirement.

According to I.Sommerville, a requirement is...
...a statement of a system service or of a system constraint

Requirements engineering is the process of establishing 3 main aspects about a system. What are they?
(1)The services a customer requires from the system; (2)The constraints under which it must operate; (3)The constraints under which it must be developed

Requirements engineering is an iterative and incremental process, that includes 3 main phases. What are they?
Elicitation, specification and validation

What is a preliminary feasibility study useful to?
To establish if the system can be developed with the given resources, time and constraints

A system stakeholder is...
Any person or organization who is affected by the system and/or has a legitimate interest in it

What is a practical way of describing the difference between user requirements and system requirements?
User requirements describe requirements from the "problem" point of view. System requirements describe them from the "solution" point of view.

In short, functional requirements are...
...statements of services the system should provide

In short, non-functional requirements are...
constraints under which the system must operate, be developed, standards it must abide to, etc.

What are two viable approaches to requirements elicitation?
Interviews (open or closed) and ethnography

What is the resulting artifact of requirements specification?

A Software Requirements Specification (SRS) document

What is requirements validation concerned with?
With demonstrating that requirements define the system that customers really want

Requirements validation must check many aspects of the collected requirements. Name at least three of them.
Validity, consistency, completeness, realism, verifiability, comprehensibility, traceability, adaptability.

In practical terms, what does a use case describe?
It describes typical and exceptional ways in which a real-world actor interacts with the system

What is the typical device to capture requirements in agile models?
User stories

UI design

The internationalization (or i18n) process of a software product has to do with...
designing and developing it so that it can be adapted to a target audience without engineering changes (supervises localization)

The localization (or l10n) process of a software product has to do with...
adapting it to a given region (underlies internationalization)

What does the principle of least surprise refer to?
To the fact that users should in principles be able to anticipate how the software component will behave

What does the principle of familiarity refer to?
To the fact that the software product should use concepts and terms familiar to the user

What does the principle of consistency refer to?
To the fact that comparable operations should be activated in comparable ways and that comparable components should have comparable formats

What does the principle of feedback refer to?
To the fact that the software should maintain a two-way communication with the user and provide visual and auditory feedback

The degree to which a software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use" is said (1 word)...
usability

What does the memory load principle refer to?
To the fact that we must reduce that amount of information that users must remember between actions

What does the principle of efficiency refer to?
To the fact that we must seek efficiency in dialogue, motion and thought

What does the principle of recoverability refer to?
To the fact the we must allow users to recover from their errors

What does the principle of user guidance refer to?
To the fact that we must incorporate some context-sensitve user guidance and assistance

What simple approach can we use for possible cultural issues arising from the design of our product?
Put users in control and allow them to customize the software product

An ASCII code position is a Unicode code position. True/false? Why?
True. The first 128 Unicode code positions correspond to those of the ASCII table.

How does usability testing differ from other testing processes?
Usability tests cannot be automated

Usability testing verifies several attributes of a software product. Name at least 3...
learnability, _efficiency_ (or _speed of operation_), _errors and recoverability_, _flexibility_, _robustness_

Software Design

In object-orientation, a class should be designed to represent... what?
what an entity IS

A sufficient implementation of a software component is one, that...
...captures the characteristics of its abstraction, that are necessary for a meaningful and efficient interaction

A complete implementation of a software component is one that...
...captures ALL the characteristics of its abstraction

A primitive implementation of a software component is one that...
...cannot be realized without access to the internal characteristics of its abstraction

Single Responsibility Principle or SRP. Define briefly.
The design principle for which software components must be developed in such a way that they respond to a single role's requirements

Open-Closed Principle or OCP. Define briefly.
The design principle for which software components must be developed so that their funcionalities may be extended without changing their implementation

Liskov Substitution Principle or LSP. Define briefly.
A design principle that refers to subtyping relations for which the nature of these relations is semantic rather than purely sintactic

Interface Segregation Principle or ISP. Define briefly.
A design principle for which no components should be forced to depend on methods they do not use, thus suggesting the definition of smaller, more focussed, interfaces

Dependency Inversion Principle or DIP. Define briefly.
A design principle for which higher level modules should be independent of the implementations of the lower level modules they use

Encapsulation. Define briefly.
A programming language device, that allows the bundling of data and the methods/functions that operate on that data

Information hiding is that design principle, that aims to hide... what exactly?
The design decisions that are most likely to change

Architectural design. Define briefly.
The process (as a series of decisions) that identifies the main structural components of a system and their relationships

The measure of the strength of the relationship of the elements in a module is called...
Cohesion

Coupling. Define briefly.
The degree of interdependence between software modules

A functional approach to system decomposition is indicated in what type of context?
In a context in which requirements are stable

An object-oriented approach to system decomposition is indicated in what type of context?
In a context in which requirements are unstable and/or evolve

Software is subject to continuous change. For this reason we want to design components, that, when modified, allow to minimize... what exactly?
The number of changes to related components

The OSI model is an example of the fundamental design principle called Separation of Concerns (SoC). What type of architecture is used?
A multi layer architecture

In object-orientation, inheritance somewhat conflicts with loose coupling. Why?
Because in a parent-child inheritance structure the changes made to the parent directly and inevitably impact the child, thus contrasting with the ideal of loosely coupled components

"Program to an interface, rather than to an implementation". Elaborate briefly.
Software components should be coupled to contracts (i.e., interfaces) rather than to concretions (i.e., implementations), so that the latter can vary and/or can be replaced freely, rendering the design more flexible and easier to evolve

Polymorphism. Define briefly.
The provision of a single _interface_ to entities of different _types_

In object-orientation, an interface should be designed to represent... what?
what an entity DOES

Unified Modeling Language (UML)

What does an activity diagram describe?
It describes a workflow: either organizational or computational.

What does a class diagram describe?
It describes the structure of a system in terms of its classes (with their properties and methods), and the relationships among them.

UML is a formal, comprehensive and standard modelling language. Why is the _formal_ aspect so important?
Because the strongly defined meaning of the UML elements avoids misunderstandings.

UML is a formal, comprehensive and standard modelling language. Why is the _standard_ aspect so important?
Because its notation system can be used consistently across all application domains.

UML is a formal, comprehensive and standard modelling language. What is meant by _comprehensive_?
It means that UML allows to represent all aspects of a system, from all the necessary points of view (i.e., use-case, logical, process, development/implementation, physical/deployment).

What does a state diagram describe?
The _states_ a system can be in and the events/actions/behaviours that determine the transation from a state to another.

What does a sequence diagram describe?
It describes the (sequence of) messages exchanged by a set of objects for the realization of a given functionality/task.

What does an use-case diagram describe?
It describes the users' interactions with the system.

UML is said to be a 4+1 (views) model. Why is the use-case view represented as a +1 overlay?
Because all the other views rely on the use-case view to guide them.

UML diagrams express one of two fundamental aspects of a system. What are they?
The _structural_ or the _behavioural_ aspect.

UML is said to be a 4+1 (views) model. In essence, what does the use case view describe?
It describes the functionalities of the system from the perspective of the outside world.
Basically: what the system is supposed to do.

UML is said to be a 4+1 (views) model. In essence, what does the logical view describe?
It describes the _abstractions_ of the system's parts. Basically: what the system is made up of and how the parts interact.

UML is said to be a 4+1 (views) model. In essence, what does the process view describe?
It describes the processes within the system. Basically: what happens within the system.

UML is said to be a 4+1 (views) model. In essence, what does the development view describe?
It describes how the system is organized into modules and components.

UML is said to be a 4+1 (views) model. In essence, what does the physical view describe?
It describes how the system (as described in the logical, process, and development views) translates into real-world entities.