

Lecture Recap Quiz 3

- Requirements elicitation:
 - A.) Defines the system in terms understood by the developer
 - **B.)** Describes the purpose of the system under construction
 - C.) Identifies the “how” of the system
- Constraints:
 - A.) Only identify the “what” of the system
 - B.) Do **not** describe Implementation technologies
 - **C.)** Are used to describe legal and operation requirements
- Techniques to describe requirements:
 - A.) Use Case describes the interaction between a specific user and the system
 - B.) Evaluation Scenario is used during system deployment
 - **C.)** User story describes a functional requirement from the perspective of the user
- Analysis model:
 - A.) is the class diagram describing the system but simplified, omitting access modifiers, attribute types and return types
 - **B.)** is the object model and the dynamic model combined
 - C.) describes the structure of the system

Lecture Recap Quiz 4

• Design Goals:

- A.) Any ^{gen}functional requirement is a design goal
- B.) Are identified with respect to design methodology, design metrics, implementation goals
- C.) Are one of the 8 issues of object design

• Subsystem Decomposition:

- A.) A ^{facade} subsystem is a group of externally visible operations provided by an interface ^{divide into smaller parts to reduce complexity} ^{do NOT describe functions WITHIN the system} ^{API goes beyond that}
- B.) Good system design when classes are loosely coupled and subsystems are related to each other via many associations _I
- C.) A subsystem is a collection of classes, associations, operations, events that are closely interrelated with each other

• Architectural styles:

- A.) A layered architecture is ^{closed if it can call operations only from the layer directly below it} closed when a layer can call operations from all layers below _{open}
- B.) Open architecture fulfil the design goals ^{high efficiency -> high coupling} portability and achieve low coupling
- C.) 3 tier architecture is a SW architecture where the 3 layers are allocated on separate HW nodes

Lecture Recap Quiz 5

• MVC:

- A.) The view subsystem sends updates to the controller subsystem and model subsystem
- B.) The difference between 3 tier architectural style and MVC is that MVC is hierarchical
- C.) Decouples data access and presentation, allowing multiple synchronized views of the same model

3-tier architectural style

if you change the view, you do not need to change other components

• Concurrency:

multiple things going on in the same time and you cannot

tell which thread will finish earlier than the others

- A.) A race condition is a design flaw where the output of a process depends on the specific sequence of other events
- B.) Logical concurrency is when threads are provided by hardware
- C.) Addresses the non functional requirements flexibility and maintainability as tasks are assigned to separate threads

software

scalability and throughput (=performance)

• HW/SW Mapping:

- A.) Addresses the question how UML models can be realised in code
- B.) Control objects are mapped to processors, entity objects to memory and boundary objects to I/O devices
- C.) The difficulty of HW/SW mapping is determining the cost of the HW needed to realise the system

• Access Control:

analysis phase

- A.) During system design we model access rights by associating use cases with the actors
- B.) Access right is the row in the access matrix and it models all possible actors that interact with the system
- C.) Capability associates a (class, operations) pair with an actor

an entry

way to implement an access control: by storing access rights somewhere where they can be looked up later or handing a certificate to a certain object which he can use to access certain other methods and objects

• Software Control:

- A.) Centralised design leads to additional communication overhead
- B.) Centralised control can be modelled using the Fork diagram
- C.) Event-driven centralised control is when control resides within the program code

central object has a info going out of it

the user triggers a process (from outside)

Lecture Recap Quiz 5

1. Reuse:

- A.) The goal of inheritance is only to achieve interface specification
- **B.)** White box reuse could be achieved using inheritance
- C.) Delegation reuses implemented functionality in the superclass

2. Generalization vs. Specialization:

- **A.)** Generalization usually leads to model transformation
- B.) Refactoring means restructuring the UML model to fill in the gap between object design and system design
- C.) In specialization inheritance we first discover the subclass then the superclass

3. Composite Pattern:

- A.) **Strategy** Composite pattern is used when we need to choose between multiple strategies during runtime
- B.) Composite pattern is a **Structural** **Creational** pattern
- **C.)** The composite pattern lets a Client treat an individual class called Leaf and Compositions of Leaf classes uniformly

4. Bridge Pattern:

- A.) The bridge pattern provides a bridge between the concrete **application domain** **implementation** and the solution domain
- **B.)** The degenerated bridge pattern has no application domain taxonomy
- C.) The bridge pattern allows to bind the interface and its subclass **at** **before** startup time of the system

5. Proxy Pattern:

- **A.)** The proxy pattern allows to defer **object creation** and **object initialization** to the time the object is needed
- B.) During access control the proxy object acts as a stand-in for an object which is expensive to create **Substitution use case**
- C.) The implementation of **request()** in Proxy uses **delegation** **inheritance** to access **request()** in RealSubject

Lecture Recap Quiz 7

- Adapter Pattern

- A.) Both the adapter and the bridge pattern delay design decisions till the startup time of the system
- B.) The adapter pattern is used in Greenfield and Reengineering projects
- ☒ C.) A legacy system is often considered as an irreplaceable system

- Observer Pattern

- A.) The observer pattern is used for systems that are expensive to create during runtime
- ☒ B.) The observer pattern is used for systems that change their state quite often and need to be consistent across multiple views when the state changes
- C.) The observer pattern is used during interface engineering

- Strategy Pattern

- A.) The strategy pattern allows us to connect two incompatible components
- B.) In the strategy pattern the client depends on the specific implementation of the underlying strategies, so if the strategy changed the client also needs to change
- ☒ C.) The strategy pattern allows to switch between different algorithms at runtime based on changing circumstances

Lecture Recap Quiz 8

- Terminology
 - A.) Error is when the observed behaviour is not equal to the expected behaviour. *failure*
 - B.) Verification is the activity of checking for deviations from the specified behaviour of the system. *Validation*
 - C.) When the system has an error, any further processing could lead to failure. *Am I building the product right?*
- Unit Testing
 - A.) Black box testing is done using analysis knowledge to discover new features. *predict input/output behavior*
 - B.) White box testing tests what is done instead of what should be done. *Am I building the right product?*
 - C.) During unit testing developers test groups of subsystems. *small integration testing*
- Integration Testing
 - A.) During integration testing, the developers test the complete system and how well all the components are integrated together.
 - B.) A driver acts as a replacement of the actual subsystem and provides the same interface.
 - C.) Later integration exposes the system to higher risks of unexpected failures, vertical integration is therefore used to build, integrate and test subsystems as early and as frequently as possible.
- System Testing
 - A.) During system testing, the complete system is examined in the target environment to see whether it meets the client's requirements or not.
 - B.) Fuzzing refers to running a program on many random, abnormal inputs to detect deviations from the expected behaviour.
 - C.) White box fuzzing generates input without any knowledge of the program.
- Model Based Testing
 - A.) The Oracle contains all data that is required for the test case execution. *predicts the output of the input data that contains all the data*
 - B.) With automated testing, it is possible to test a system completely. *u can never fully test the system*
 - C.) It's helpful to separate developers from testers. *developers have biases about the things they have developed*
- Object Oriented Testing
 - A.) The replacements for collaborators are mock objects.
 - B.) The System Model contains all replacements of the collaborators. *and SUT*
 - C.) The SUT gets isolated to prevent interferences or allow early testing. *system under test*

Lecture Recap Quiz 9

• Software Development as Application Domain

- A.) The activity entity centered views of SW development lifecycles displays software development as a set of activities deliverables.
- ☒ B.) Software development lifecycles should be tailored by naming, cutting and ordering the activities to fit the current project.
a process consists of a set of activities
- C.) A process group consists of activities.

• Linear Models

- A.) Software development is a linear process, therefore managers love sequential models.
- B.) In the V-model development and testing are performed in parallel.
non-linear model and nothing is done in parallel
- ☒ C.) In the waterfall model you cannot turn back once an activity is completed.

• Iterative Models

- A.) The unified process model consist of 4 phases, each phase has one iteration.
several
- B.) The spiral model can deal with change within an activity.
not exactly because if there is a change WITHIN activity, this activity needs to be redone (changed) later iterative development means to re do something later that is already done
- ☒ C.) In an incremental development various parts of the system are developed at different times and integrated as they are complete.

• Agile Models

- ☒ A.) The main purpose of the Scrum daily is to reduce risk, through early information sharing.
- B.) In Kanban we start by changing the existing process to a completely new agile process.
is incrementally changed adapts to changes and sees them as an opportunity
- C.) An empirical process avoids deviation.
defined process control avoids deviation

Lecture Recap Quiz 10

- Software configuration management
 - A.) SCM manages the whole project analysis phase of a software project
 - B.) Promotion is a new version internally available for developers
 - C.) Configuration items are identified during requirements elicitation
- Change management
 - A.) A version is changed by implementing new bug fixes → change request process (includes much more than a)
 - B.) Change policy lists criteria a promotion or release has to conform to and c)
 - C.) Change management just handles the naming and creation of a new version ↗
- Version control systems
 - A.) A distributed VCS does not enable developers to work offline
 - B.) A VCS allows multiple developers to view the same file simultaneously sequentially through a lock token
 - C.) A merge conflict occurs if multiple developers modify the same part of the same file
- Branch management
 - A.) The development feature branch is used to develop new code to solve a certain issue. internal promotions
 - B.) A review tries to make sure that no faulty code gets promoted
 - C.) To prevent merge conflicts one should pull from feature development branch
- Continuous integration
 - A.) A disadvantage is having too many executable versions of the system does not matter how many executable versions of the system you have, you delete them after anyways
 - B.) In regression testing executing large test suits after each change is easy often quite costly done and finds many errors immediately
 - C.) Build Agents are used to run build processes in parallel
- Release management
 - A.) A release manager plays the role of a liaison between different business units to achieve timely delivery of software products
 - B.) The software project should only be delivered after the implementation of everything was completely done release executable versions as often and early as you can
 - C.) Release management speeds up slows down publishing the software product to the market because it goes through many cycles first before release

Lecture Recap Quiz 11

• Software quality management

- A.) Quality means a product should implement the functional requirements elicited by developers at the beginning of the project
also standards and implicit requirements not only by developers
- B.) Agile quality planning identifies important release dates and quality goals
less emphasis on strict deadlines and goals
- C.) Quality control is done by the client to test if their acceptance criteria are met or not
and developers

• Software Reviews

- A.) The purpose of code reviews is to check the implementation progress of the requirements requested by the client
verification => uncover errors at various phases of the project
- B.) A walkthrough is a planned code review
informal
- C.) In analysis review the developers check whether the requirements specification is clear and realistic

• Static vs. Dynamic Analysis

- A.) Static analysis only catches a small number of quality analysis problems
Dynamic
- B.) Static analysis tests the implementation of subsystems
static -> input/output behaviour
- C.) Dynamic analysis catches code duplications