

Naam:

Belangrijk: Schrijf je antwoorden kort en bondig in de daartoe voorziene velden. Elke theorie-vraag staat op 2 punten (totaal op 24). De oefening staan in totaal op 16 punten. Het geheel staat op 40 punten.

Vraag 1. Introduction [..../2]

Geef de 6 principes van Extreme Programming.

- Fine scale feedback
- Continuous process
- Shared Understanding
- Programmer welfare
- Coding -> first unit test then code
- Testing -> all code must have tests, acceptance tests are run often

Vraag 2. Project Management [..../2]

Wat is een “known known”, een “unknown known” en een “unknown unknown”?

Known knowns: The things you know you know, you can safely make assumptions here.

Unknown knowns: The things you know, you don't know. You can prepare for these during planning.

Unknown unknowns: The things you do not know, you don't know. These you cannot prepare for during planning. The best you can do is being aware and spot opportunities and do a thorough risk analysis.

Vraag 3. Use Cases [..../2]

Geef twee voor- en twee nadelen van het gebruik van Use Cases.

Voordelen:

- requirements more understandable (actors provide end users perspective)
- requirements more precise. (scenarios are sufficiently detailed to test)
- requirements open. (Actors perspective emphasizes the what (and much less the how)
- Helps to validate solution against requirements
- Helps to verify the requirements against users needs

Nadelen:

- requires close interaction with various stakeholders
- needs iterations to improve earlier misconceptions
- a lot of hard work
- Use cases tend to result in hard to maintain systems
- Identifying actors and use cases may omit requirements (Completeness not guaranteed)
- Focus on scenarios restricts evolving requirements.

Naam:

Vraag 4. Domain Models [..../2]

Kan je uitleggen hoe “role-playing” werkt? Denk je dat dit helpt bij creatief denken?

- Role-playing is a way to achieve common understanding between all parties involve(domain experts, analysts, ...). It helps in creative thinking due to non-strict setting of the role-playing and the interaction between individuals.

- Basic Steps

- Create list of scenarios
- Assign Roles
- Each member receives a number of CRC Cards
- Repeat
 - Rehearse Scenarios + Script = Responsibilities on CRC Cards
 - 4.2 Correct CRC Cards and revise scenarios + Rehearsals will make clear which parts are confusing
 - Until scenarios are clear
- Perform final scenario

Vraag 5. Testing [..../2]

Wat is het verschil tussen stress-testing en performance testing?

Stress-Testing: Tests extreme conditions - tries to break the system.

Performance testing: Test run-time performance in normal conditions, time consumption memory consumption. Checks if the system is performant enough according to the requirements.....

Vraag 6. Design by Contract [..../2]

Wat is het *Liskov substitutie principe*?

Replacing an instance of a superclass by each of its subclasses is allowed.

.....

Waarom speelt dit een belangrijk rol binnen het object georiënteerde paradigma en voor design contracten?

It defines the rules for inheritance and therefore subcontracts.

Vraag 7. Formal Specifications [..../2]

Wat heeft “cleanroom development” te maken met Formele Specificaties?

Cleanroom development tries to reduce cost and time by eliminating errors in specification and design and by ensuring that the implementation does not introduce defects. This can be done with formal specification and verification: assertions, pre and post conditions.....

Naam:

Vraag 8. Software Architecture [..../2]

Wat is een component? En wat is een connector?

Component: An encapsulated part of a software system with a designated interface. Components may be represented as modules (packages), classes, objects or a set of related functions.....

Connector: A connection between components.

There are static connectors that appear directly in source code (e.g., use or import keywords) and dynamic connectors that deal with temporal connections (e.g., method invocations).....

Vraag 9. Quality Control [..../2]

Hoe zou je een review meeting organiseren?

Review meetings should:

- typically involve 3-5 people
- require a maximum of 2 hours advance preparation
- reviewers use checklists to evaluate products
- last less than 2 hours

The review minutes should summarize:

- What was reviewed
- Who reviewed it?
- What were the findings and conclusions?
- Decision
 - Accepted without modification
 - Provisionally accepted, subject to corrections (no follow-up review)
 - Rejected, subject to corrections and follow-up review

Vraag 10. Software Metrics [..../2]

Wanneer voldoet een coupling metriek niet aan de “representation condition”?

Geef ook een voorbeeld.

If the metric doesn't measure what we understand as coupling.

vb1: high LCOM value (high cohesion) by accessor methods

vb2: classes with low coupling but nevertheless high CBO value, because there is no difference between data, method or inheritance coupling

Vraag 11. Refactoring [..../2]

Geef 4 code smells die met refactoring kunnen opgelost worden.

- Duplicated code
- Nested conditionals
- Large classes/methods

Naam:

- Abusive inheritance

Vraag 12. Conclusion [.../2]

Als je het “*Silver Bullet*” artikel hebt gelezen:

Wat is het verschil tussen essentie (essence) en ongelukken (accidents)?

- essence: the difficulties inherent in the nature of software
- accidents: those difficulties that today attend its production but are not inherent.

Als je het “*Killer Robot*” artikel hebt gelezen:

Geef de drie belangrijkste redenen waarom het fout ging en leg uit waarom je dit de belangrijkste vindt.

- Developers of Killer Robot worked under enormous stress => bad planning
- Killer Robot Programma was prima Donna, co-workers claim => makes verification impossible
- Silicon Techtronics Employee Admits Faking Software Tests

Naam: