

Oefentoets 2 Dev - C4, Continuous Documentation, RabbitMQ en Micro front-ends

C4 Model

1. Hoe visualiseert het C4 model de architectuur?
 - a. Het toont het systeem of systeemelementen op verschillende niveaus, met communicatielijnen onderling en met externe actoren (gebruikers of systemen)
 - b. Het visualiseert met een 3D weergave alle 'quality attributes' van de architectuur doordat de gebruiker er doorheen kan in- en uitzoomen (animatie, e.g. 4D)
 - c. Het toont de Controle structuren, de Cardinaliteit en de Code Checkpoints van een softwaresysteem
 - d. Het toont de interne communicatiepatronen van een systeem
2. Welke stelling over *Code* in het C4 model is waar?
 - a. Het C4 model is niet bedoeld voor code niveau details, omdat deze taal afhankelijk zijn; C4 is alleen voor hogere niveau documentatie.
 - b. Het C4 model bevat speciale 'code diagrammen', die compliant zijn met de ISO65001 standaard.
 - c. Code diagrammen zijn optioneel in C4, maar je kunt elk UML diagram toevoegen om details van de onderliggende code van een component te geven/ontwerpen.
 - d. Code diagrammen zijn essentieel in C4, je gebruikt flow diagrammen voor het modelleren van sequence, iteratie an conditionele logica van de code in Containers.
3. Hoe ziet een *Context Diagram* eruit in het C4 model?
 - a. Dit is het hoogste niveau diagram dat de systeem context toont zonder details van de interne structuur.
 - b. Dit is een korte 'high level' beschrijving gericht op business stakeholders om hen de initiële context te geven van een nieuw project
 - c. Dit is het laagste niveau diagram, dat alle details toont van het systeem functionerend in zijn context zodat business stakeholders interne use cases kunnen valideren.
 - d. Dit is een uitgebreide beschrijving gericht op developers om hun context te geven van het bedrijfs perspectief op het systeem.
4. Wat is een goede beschrijving van het C4 model?
 - a. Documentatie door designers, voor designers!
 - b. Documentatie door designers, voor developers
 - c. Documentatie door developers, voor developers!
 - d. Documentatie door developers, voor designers

RabbitMQ

5. Welk product is grotendeels vergelijkbaar met RabbitMQ?
 - a. SoapUI
 - b. Blockchain
 - c. Kerberos
 - d. ZeroMQ
6. Welk concept is een 'message pattern' dat RabbitMQ ondersteund?
 - a. Work queue
 - b. Pipe and Filter
 - c. Programmable Contracts
 - d. Prototype
7. Welk antwoord laat de manieren van 'message publishing' zien van langzaam naar snel?
 - a. No Guarantees, Persisted Messages, HA QUeues, Transactions
 - b. Persisted Messages, HA Queues, Transactions, No Guarantees
 - c. No Guarantees, Transactions, HA QUeues, Persisted Messages
 - d. Persisted Messages, Transactions, HA Queues, No Guarantees
8. Welk stelling is *CORRECT*? Er is slechts 1 antwoord mogelijk.
 - a. Camel biedt implementatie van enterprise integration patterns.
 - b. Camel kan gegevens van en naar queues lezen en schrijven maar niet van en naar andere bronnen.
 - c. Camel ondersteunt alleen XML als dataformaat voor het versturen van berichten.
 - d. Camel ondersteunt van alle beschikbare message brokers alleen RabbitMQ.
9. Waaruit bestaat een 'message' in een messaging systeem typisch?
 - a. Een ontvanger en een verzender
 - b. Een header en een body
 - c. Een 'message initiator' en een medium
 - d. Een publisher en een subscriber
10. Wat is een eigenschap van een *queue* qua volgorde van berichten die je erin stopt?
 - a. Last in, first out
 - b. First in, first out
 - c. Fast in, fast out
 - d. Kan allemaal, hangt af van type exchange dat queue implementeert

Continuous documentation

1. Zet de 3 fasen in een project (Continuous Documentation volgens
 - a. Knowledge acquisition, Knowledge building, Knowledge transfer
 - b. Knowledge building, Knowledge acquisition, Knowledge transfer
 - c. Knowledge building, Knowledge transfer, Knowledge acquisition
 - d. Knowledge transfer, Knowledge acquisition, Knowledge building

Micro front-ends

12. Welke van onderstaande is de beste reden om een micro-frontend te implementeren?
 - a. Het oplossen van de chaos in een frontend-codebase door de code anders structureren
 - b. Als je veel verschillende teams hebt die aan hetzelfde onderdeel van de frontend werken
 - c. Als je veel verschillende teams hebt die elk aan een ander onderdeel van de front-end werken
 - d. Als je een “horizontale” werkverdeling hebt bijv: Styling Team A, Formulier code Team B etc.
13. Welke van de volgende uitspraken over micro-frontends is correct?
 - a. Micro-frontends moeten in SolidJS worden geschreven.
 - b. Micro-frontends kunnen worden geschreven in verschillende frameworks, afhankelijk van de vereisten en integratiestrategieën.
 - c. Micro-frontends kunnen alleen met WebPack worden geïntegreerd.
 - d. Micro-frontends vereisen dat alle componenten in Vanilla JavaScript worden geschreven.

Einde toets

Goede antwoorden:

1A, 2C, 3A, 4C, 5D, 6A, 7D, 8A, 9B, 10B, 11A, 12C, 13B