Prüfungsfragen SE Examia:

|  |  |  |
| --- | --- | --- |
| Nr. | Was | Wo (Kapitel) |
| 1 | Kapselung / Information Hiding | Kap. 6 / S.123 |
| 2 | Open-close-Prinzip | Kap. 6 / S.87 |
| 3 | L-Substitutions-Prinzip | Kap. 6 / S.136 |
| 4 | Function point /aufwandsschätzung | Kap. 4.1 |
| 5 | Schichten nach Starke !!!wichtig!!! | Kap 6 / S. 28 |
| 6 | SCRUM | Selber |
| 7 | Anforderungen an SW (funktional und nicht-fkt) | Kap. 4 /S.33+wiki |
| 8 | Zerbrechlichkeit | Kap. 6 / S.70 |
| 9 | Codeinspektion <-> Test | Kap 8 / S. 12 |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |

Wichtige Themen

* Lasten/pflichtenheft
* Requir. Engi.
* Aktivitätsdiagramm theorie genau

https://drive.google.com/folderview/id=1w9HvvelW-pVnbJ-5SSBuJ207hUAWGPfM

anforderungen an sw(funkt&nich funkt): kap 4 folie 33 + wiki

1.funktional:

bsp:Wie verhält es sich in bestimmten situationen,was soll das produkt tun,reaktionen auf eingaben,welche dienste stellt es zur verfügung

2.wie gut soll es das tun

bsp:Wie schnell arbeitet es,wie viel speicher braucht es,wie gut ist es benutzbar,auf welche systeme is es übertragbar?

Function point/aufwandsschätzung: Kap4.1

Expertenschätzung: Fachleute nutzen erfahrung

Algorithmische Schätzung: Kosten werden aus bekannten oder leichter und genauer geschätzten Größen berechnet.

Function point Anforderungen werden kategorisiert und bewertet und daraus aufwand berechnet.

!WICHTIG!schichten nach starke: kap 6 folie 26

Warum?:Darstellung oft sehr komplex,sichten verinfachen durch aufteilung.Meist brauchen Projektbeteiligte nur wissen über teile des systems.

Kontextsicht:Umgebung des Systems,Nahcbarsysteme(UML:Paket Komponenten Klassen)

Bausteinsicht:System innerhalb,statische strukturen(UML:Paket Komponenten Klassen)

Laufzeitsicht:Verhalten des Systems und der Bausteine zur laufzeit (UML:Aktivität,Sequenz,Kommunikation)

Verteilungssicht:Hardwarekomponenten des Systems Umgebung(Prozessoren Betirebssystem etc)

zerbrechlichkeit:kap6 folie 70

kleine fehler lösen fehler in anderen sw teilen aus

Symptom:großteil der zeit in sw projekten ist bugfixing

open/closed: Kap 6 folie 87

Lösung:Komponenten sollen offen für erweiterungen und geschlossen für veränderungen sein.

Hinzufügen einer neuen eigenschaft ist ohne änderung am code möglich.

capselung/information hiding:Kap 6 Folie 123

Kapselung=Attribute und methoden auf attribute sinnvoll zusammengefügt(in einer klasse).

Information Hiding:innere Struktur geht clients nichts an und änderungen betreffen sie nichts(private getter setter).

L-Substitutionsprinzip: Kap6 folie 136

Lösung:Klassen sollen durch unterklassen ersetzbar sein

!WICHTIG!scrum ->wikipedia (nichts dazu auf folien gefunden!)

3 rollen und deren aufgaben

1.Product owner(erstellt,priorisiert und erläutert Produkteigenschaften)

2.Entwicklungsteam(umsetzung der Produktfunktionalitäten)

3.scrum master(kümmert sich drum dass scrum funktioniert und kontrolliert einhaltung)

3 fragen im daily scrum:

1.was erreicht seit letztem ds

2.was erreichen bis nächsten ds

3.welche Herausforderungen sind aufgetreten

4 wesentliche meetings scrum projekt

1.Sprint-Planungstreffen

2.Daily Scrum

3.Sprint Review

4.Sprint Retrospektive

Codeinspenktion <->test:kap8 s 12

Vorteile der Inspektion:

1.Qualität kann geprüft werden

2.kann bei nicht fertigem code gemacht werden.

3.Bei test können fehler andere fehler überdecken