Kritikus beágyazott rendszerek

Vizsgakérdések

1. „General safety concepts” (általános biztonsági koncepciók) témakör:

1.1. Mit jelent a baleset, a veszély, a kockázat, a biztonság és a funkcionális biztonság fogalma és milyen kapcsolatok vannak ezen fogalmak között?

1.2. Milyen lépései vannak a kockázatanalízisnek?

1.3. Mit jelent a *biztonságintegritás*?

1.4. Milyen használati módjai lehetnek biztonságkritikus rendszereknek, és hogyan adhatjuk meg az egyes használati módok esetén a biztonságintegritási elvárásokat?

1.5. Hogyan értelmezzük a *biztonságintegritási szinteket*?

1.6. Milyen általános struktúrája van a követelményspecifikációnak biztonságkritikus rendszerek esetén?

1.7. Milyen fázisai vannak egy általános biztonsági életciklusnak?

1.8. Mi a feladata a biztonsági szervezetnek a fejlesztés során?

2. „Safety Requirement Specification” (biztonsági követelményspecifikáció) témakör:

2.1. Hogyan értelmezzük a *kockázatcsökkentés* fogalmát (EUC kockázat, elfogadható kockázat és maradékkockázat)? Milyen az alapjául szolgáló rendszermodell?

2.2. Milyen a biztonsági követelmények két fő típusa és hogyan néz ki a követelmények hierarchiája? Mi a biztonsági követelmények szerepe a biztonsági funkciók kialakításában és hol helyezkednek el a kapcsolódó lépések a biztonsági életciklusban?

2.3. Mik a kockázatfelmérés lépései? Hogyan mérjük fel a veszélyes események gyakoriságát és következményeit? Az elfogadható kockázatra vonatkozó kritériumoknak milyen fajtái vannak? Rajzoljon fel egy tipikus kockázati sáv diagramot és értelmezze!

2.4. A kockázatcsökkentés szükséges mértékének meghatározása. Milyen rendszerekkel érjük el a kockázatcsökkentést? Mi az ALARP elv és mekkora kockázatcsökkentést ír elő? Hogyan tükröződik az ALARP elv az IEC 61508 szabványban (ábra, táblázat)?

2.5. Hogyan történik a biztonságintegritási szint számítása egy biztonsági rendszerre, ha a következmény állandó?

2.6. Mi a *kockázati gráf*, milyen paraméterektől függ, hogyan történik a segítségével a szükséges SIL szint meghatározása? Mik ennek a módszernek a gyenge pontjai?

2.7. Mi a *veszélyes esemény súlyossági mátrix*, milyen paraméterektől függ, hogyan történik a segítségével a szükséges SIL szint meghatározása? Mik az alkalmazásának feltételei?

3. „Hardware Safety Integrity” (hardver biztonságintegritás) témakör:

3.1. Mi a *biztonságos hibahányad* (SFF)? Hogyan értelmezzük a hibatűrés mértékét? Mi az alrendszerek A és B típusa, és milyen architekturális követelmények vonatkoznak adott SIL érték elérésére?

3.2. Hogyan határozzuk meg egy több csatornás, több alrendszerből álló rendszer SIL szintjét?

3.3. Mik a *közös okú hibák* (CCF)? Hogyan vesszük figyelembe a közös okú hibákat (valószínűség) és hogyan csökkenthetjük az ilyen hibák előfordulásának valószínűségét?

3.4. Mi az *igényre bekövetkező hiba átlagos valószínűsége* (PFD_{avg})? Hogyan számítjuk, mik a meghatározásának főbb lépései? Mik a módszer alkalmazásának főbb feltételei?

3.5. Milyen redundáns architektúrákra ad PFD_{avg} számítási módszert az IEC 61508?

4. „Software in safety-critical systems" (szoftver biztonságkritikus rendszerekben) témakör:

4.1. Milyen alapelvei vannak a szoftver-biztonságintegritási szint meghatározásának?

4.2. Hogyan kapcsolódik össze a szoftver és a hardver biztonsági életciklus?

4.3. Milyen módon határozzák meg a szabványok a szoftver fejlesztés során használandó módszereket és technikákat?

4.4. Milyen elvárások vannak a fejlesztőeszközökkel szemben?

4.5. Hogyan tehető a C és a C++ programozási nyelv biztonságosabbá?

4.6. Milyen elvárások vannak a biztonságkritikus rendszerekben használandó operációs rendszerekkel szemben?

4.7. Mik az általános elvárások a fejlesztési folyamat dokumentációjával szemben?

4.8. Milyen elvárások vannak a fejlesztői szerepkörök függetlenségével szemben?

5. „Embedded Safety-Critical Systems in the Nuclear Industry" témakör:

5.1. Mik a nukleáris alkalmazási környezet sajátosságai? Mik a nukleáris biztonság főbb célkitűzései?

5.2. Melyek a nukleáris biztonság alapelvei? Mi az 5 legfontosabb mérnöki (biztonsági) korlát? Mi a mélységi védelem elve, és mi az 5 tipikus mélységi védelmi szint célja és megvalósítási módja?

5.3. Mik a *feltételezett kiváltó események* (PIE), milyen fajtáik vannak, milyen forrásból származnak?

5.4. Milyen üzemállapotokat, üzemi tranzienseket és üzemzavarokat különböztetünk meg súlyosság alapján egy nukleáris létesítményben? Mi ezek bekövetkezési valószínűsége, mi az elfogadott kockázat?

6. „Standards in Avionics System Development" témakör:

6.1. Mutassa be általánosan a DO-178B szabványt és ismertesse legfontosabb alapelveit!

6.2. Hogyan kapcsolódik a teljes rendszer fejlesztése a szoftverfejlesztési folyamatokhoz?

6.3. Ismertesse a DO-178B-ben definiált SW fejlesztési folyamatot!

6.4. Ismertesse a DO-178B-ben definiált verifikációs folyamatot és részletezze a tesztelésre vonatkozó részeit!

6.5. Miben hasonlít, és miben különbözik a DO-178C az azt megelőző DO-178B szabványtól?

7. „Safety Case" (biztonsági ügy) témakör:

7.1. Mi a célja és mit kell tartalmaznia a *biztonsági ügynek* (safety case)?

7.2. Mit kell bemutatni a biztonsági ügy műszaki részében?

7.3. Milyen elemek és relációk vannak a Goal Structured Notation (GSN) jelölésrendszerben?

7.4. Mi az általános struktúrája a GSN segítségével felépített érvelésnek?

7.5. Szoftver esetén hogyan épülhet fel az érvelés?

7.6. Hogyan használhatók érvelési minták a GSN segítségével?

7.7. Hogyan lehet moduláris biztonsági érvelést felépíteni?

7.8. Mik a GSN használatának előnyei és korlátai?

8. SysML & MARTE témakör:

8.1. Mi a modellvezérelt fejlesztés sikeres alkalmazásának két alappillére?
8.2. Milyen aspektusai vannak a MARTE modellezési nyelvnek és hogyan lehet a MARTE specifikációt használni?
8.3. Hasonlítsa össze az UML és SysML modellezési nyelveket!
8.4. Milyen modellezési lehetőségeket nyújt a SysML nyelv?
8.5. Milyen lépéseket javasol a SYSMOD módszertan a fejlesztés során a követelmény-analízis fázisra és a SysML mely nyelvi elemeit használja az egyes lépések során?
8.6. Milyen lépéseket javasol a SYSMOD módszertan a fejlesztés során a tervezési fázisra és a SysML mely nyelvi elemeit használja az egyes lépések során?

9. „Architecture Description Languages" témakör:

9.1. Mi a különbség az *architektúra* és a *design* között?
9.2. Milyen két architektúra leíró megközelítés terjedt el és miben különböznek?
9.3. Ismertesse a Rapide ADL nyelvet!
9.4. Általánosan ismertesse az AADL nyelvet és részletesen mutassa be a szoftver modellezésre használt elemkészletét!
9.5. Általánosan ismertesse az AADL nyelvet és részletesen mutassa be a hardver modellezésre használt elemkészletét!
9.6. Helyezze el az EAST-ADL nyelvet a járműipari fejlesztési folyamatban!
9.7. Hol és milyen módon kapcsolódik az EAST-ADL nyelv az AUTOSAR koncepcióhoz?

10. „Standardized Runtime platforms and component integration" témakör:

10.1. Ismertesse az Autosar magas szintű fejlesztési folyamatát!
10.2. Autosar esetén mi a komponens fogalma és hogyan használják modellezésre?
10.3. Ismertesse az Autosar runtime felépítését!
10.4. Ismertesse az ARINC 653 koncepcióját!
10.5. ARINC 653 esetén, hogyan működik a particionálás és a kommunikáció?
10.6. Mi az a Health Monitor és mire szolgál?

11. „Program source code generation" (program forráskód generálás) témakör:

11.1. Mi a feladata a kódgenerátornak modellalapú fejlesztés esetén? Hogyan vehetők figyelembe a platform specialitásai?
11.2. Milyen jellegzetes kiterjesztések válhatnak szükségessé, ha biztonságkritikus rendszerekhez történik forráskód generálás?
11.3. Mi a Quantum Programming (QP) módszer alapötlete?
11.4. Mi az általános struktúrája egy állapotkezelő (state handler) függvénynek QP esetén?
11.5. Milyen problémákkal kell szembenéznünk, ha Java nyelvet szeretnénk használni biztonságkritikus rendszerekben?
11.6. Milyen alapelemei vannak a Real-Time Specification for Java (JSR-1) megközelítésnek?
11.7. Mik a memóriakezelési és ütemezési megoldásai a Safety-Critical Java (JSR-302) javaslatnak?
11.8. Mit jelentenek az egyes megfelelőségi szintek (compliance levels) a Safety-Critical Java (JSR-302) javaslat esetén?

12. „Testing in mobile distributed systems” témakör:

12.1. Milyen kihívásokat jelent a mobil elosztott rendszerek tesztelése?
12.2. Milyen részekből állhat egy teszt platform, amit mobil elosztott rendszerek esetén használunk (milyen tesztelés szempontjából funkcionalitást kell bele)?
12.3. Milyen kiegészítések kellhetnek a meglévő forgatókönyv-leíró nyelvekbe ahhoz, hogy mobil elosztott rendszerek esetén is tudjuk azokat kényelmesen használni?
12.4. Milyen feladatokat kell megoldani, ha egy forgatókönyvként leírt követelményspecifikáció teljesülését akarjuk ellenőrizni egy konkrét futáson (trace)?

13. Webszolgáltatás platformok témakör:

13.1. Mik az alapvető webszolgáltatás szabványok?
13.2. Adjon meg legalább 3 nemfunkcionális követelményt, melyet az alapvető WS szabványok nem fednek le. Hogyan lehet ezeket támogatni?
13.3. Milyen főbb hibák léphetnek fel webszolgáltatás alapú rendszerekben? (Mikor/milyen jellegű hibák fordulhatnak elő?)
13.4. Milyen főbb támadástípusokkal érdemes tesztelni egy webszolgáltatás megbízható, biztonságos működését?
13.5. Milyen céllal vizsgálhatunk webszolgáltatás alapú rendszereket?
13.6. Mik a tipikus analízis kérdések?
13.7. Milyen előnyei vannak a modell alapú fejlesztésnek webszolgáltatások esetében?