



HÉT	TANANYAG	SEGÉDANYAG/MEGJEGYZÉS
A szoftvertesztelés alapjai		
1	Bevezetés, a tantárgyi követelmények és a számonkérés rendjének ismertetése	
2	A szoftver életciklusa, a szoftvertesztelés helye az életciklusban, A szoftvertesztelés célja.	
3	Tesztelési szintek, tesztpiramis	
Szoftverfejlesztési módszertanok		
4	A szoftverfejlesztési módszertanok bemutatása	
5	A szoftvertesztelés helye és szerepe különböző módszertanokban	
6	A szoftvertesztelés helye és szerepe különböző módszertanokban	
7	Számonkérés	
Szoftvertesztelési módszerek		
8	Szoftvertesztelési módszerek áttekintése	
9	Doboz módszer	
10	Debugging	
11	Gyakorlati feladat: alkalmazás hibáinak feltárása	
12	Rendszer szintű tesztelés	
13	User Acceptance Testing	
14	Biztonsági és terhelési tesztelés	
15	Test-driven Development	
16	Unit tesztelés	
17	Gyakorlati feladat: Unit tesztek	
18	Év értékelése	



HÉT	SZÁMONKÉRÉS TÉMÁJA	FORMA	MEGJEGYZÉS
7	Szoftvertesztelés alapjai, szoftvertesztelés különböző módszertanokban	Írásbeli témazáró dolgozat	
11	Debugging	Gyakorlati feladat	
17	Unit tesztelés	Gyakorlati feladat	

**Szoftvertesztelés tantárgy****72 óra****A tantárgy tanításának fő célja**

A tantárgy célja, hogy bemutassa a modern szoftvertesztelés elméleti alapjait. Megmutassa a diákoknak, milyen fejlesztési módszertanok használatosak az iparban, és milyen eljárásokkal biztosítják az elkészült szoftverek minőségét.

A tantárgy oktatása során az elméleti anyag tanítása közvetlenül a gyakorlati példákon keresztül valósulhat meg, az elméleti anyag kisebb részekben történő ismertetése és annak azonnali, gyakorlati példákon, feladatokon történő gyakorlása során.

Az elméleti rész időszükséglete: 30%; a gyakorlati rész időszükséglete: 70%.

A tantárgyat oktató végzettségére, szakképesítésére, munkatapasztalatára vonatkozó speciális elvárások

—

Kapcsolódó közismereti, szakmai tartalmak

—

A képzés órakeretének legalább 70%-át gyakorlati helyszínen (tanműhely, üzem stb.) kell lebonyolítani.



A tantárgy oktatása során fejlesztendő kompetenciák

Készségek, képességek	Ismeretek	Önállóság és felelősség mértéke	Elvart viselkedésmódok, attitűdök	Általános és szakmához kötődő digitális kompetenciák
Felismeri és megnevezi a szoftverfejlesztési módszertanokat. Célhoz és elvárásokhoz illeszkedő szoftverfejlesztési módszertant választ ki.	A különböző szoftverfejlesztési módszertanok ismerete A helyzethez alkalmazkodó felhasználás módjainak megértése	Instrukció alapján részben önállóan	Legyen fogékony az információk befogadására és alkalmazására. Érdeklődjön az adott téma iránt.	
Célhoz és elvárásokhoz illeszkedő szoftverfejlesztési módszertant alkalmaz.	A különböző szoftverfejlesztési módszertanok alapelemeinek, folyamatainak ismerete			
A szoftverfejlesztés tesztelési szakaszában felismeri az Integration Level Testing (ILT) szükségességét, és alkalmazza is.	Integration Level Testing elemeinek, módszereinek, lehetőségeinek ismerete			Tesztelési környezet használata Hatékony internetes keresés és részvétel a legfontosabb szakmai közösségi platformokon (pl. GitHub, Stack Overflow) tudásszerzés és megosztás céljából
A szoftverfejlesztés tesztelési szakaszában felismeri a System Level Testing (SLT) szükségességét, és alkalmazza is.	System Level Testing elemeinek, módszereinek, lehetőségeinek ismerete			
User Acceptance Testinget alkalmaz szükség szerint.	User Acceptance Testing feltételeinek, módszereinek, lehetőségeinek ismerete			
Biztonsági tesztek alkalmaz.	Biztonsági tesztek módszereinek, lehetőségeinek ismerete			
Teljesítmény- és terhelési tesztet alkalmaz.	Teljesítmény- és terhelési tesztek lehetőségeinek, módszereinek ismerete			
TDD-t alkalmaz.	TDD lehetőségeinek, módszereinek ismerete			



A tantárgy témakörei

A szoftvertesztelés alapjai

A témakör célja, hogy bemutassa a szoftvertesztelés folyamatának elméleti alapjait, és megmutassa, hogy miért van szükség tesztelésre. A témakör megmutatja a tesztelés helyét a szoftver életciklusában.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

- Tesztpiramis (egység, integrációs, rendszer, átvételi)
- A szoftver életciklusa

Szoftverfejlesztési módszertanok

A témakör célja, hogy bemutassa a szoftverfejlesztési módszertanokat, és azokban a szoftvertesztelés helyét és szerepét.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor.

- Szoftverfejlesztési módszertanok
- V-modell
- Prototípus modell
- Iteratív és inkrementális módszertanok
- Gyors alkalmazásfejlesztés – RAD
- Agilis szoftverfejlesztés
- Scrum

Szoftvertesztelési módszerek

A témakör célja, hogy bemutassa és megértesse a diákokkal a különböző szoftvertesztelési módszereket, azok helyét, szerepét, a különböző módszerek közötti szemléletbeli eltéréseket, valamint a módszerek közötti választások szempontjait.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor.

- Integration Level Testing (ILT)
- System Level Testing (SLT)
- User Acceptance Testing (UAT)
- Biztonsági tesztelés
- Teljesítmény- és terheléstesztelés
- TDD

Unit tesztelés

A témakör célja, hogy megismertesse a tanulókkal a unit tesztelés alapjait. Ezen tudást gyakorlatok segítségével sajátítják el, egyszerű metódusok és osztályok működésének verifikálását végzik unit tesztek írásával.



A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor.

- Képes felsorolni a szoftvertesztelés szintjeit, és elmondani a szerepüket:
 - Unit teszt
 - Modul vagy komponens teszt
 - Integrációs teszt
 - Rendszerteszt
 - Átvételi teszt
- Képes C# vagy Java környezetben unit test case-eket létrehozni és futtatni őket:
 - Java: Junit, @Test, @Suite
 - C#: NUnit, [Test], [TestFixture]
- Képes a test case-ekben assertiót használni:
 - Java: assertEquals, assertTrue, assertFalse, assertNull, assertNotNull, assertSame, assertNotSame, assertEquals
 - C#: AreEqual, AreNotEqual, AreSame, AreNotSame, IsNull, IsNotNull, IsTrue, IsFalse
- Képes elmagyarázni az AAA mintát és követni unit test case-ek létrehozásakor
- Képes a unit test framework „felépítő” és „lebontó” funkcionálisát használni:
 - Java: @Before, @After
 - C#: [SetUp], [TearDown]
- Képes egyszerű külső függőség nélküli (pl.: file-olvasás, console-ra írás) osztály, függvény vagy metódus teljes működését lefedni unit tesztekkel, képes kitalálni az összes lehetséges bemenet-kimenet párost, ami szükséges ahhoz, hogy a tesztek futtatásakor a tesztelendő kód minden eleme lefusson (full line and branch coverage)