### 4. Kódminőség és alacsony szintű tesztlefedettség értékelése

#### Szolgáltatáselem célja

A szolgáltatás célja a szoftvertermék kódminőségének és alacsony szintű tesztlefedettségének értékelése elemző kollégák által készített, illetve az előzetesen verifikált és validált eszközöket használó megrendelőtől vagy fejlesztőtől kapott vizsgálati jegyzőkönyvek felhasználásával.

#### Mi nem célja a szolgáltatáselemnek?

A szolgáltatásnak nem célja a vizsgálat előkészítése és végrehajtása, a kódminőség vizsgálata során feltárt hibák javítása, illetve alacsony szintű tesztek tervezése és készítése a szoftvertermék forráskódjához.

#### Értékelés kimenetelei

A kódminőség és az alacsony szintű tesztlefedettség értékelését a szoftverterméken előzetesen végrehajtott vizsgálati eredmények alapján végezzük. Az elvárásokat a Termékkövetelmények tartalmazzák.

#### Lehetséges kimenetek

**Megfelelt**

A szoftvertermék kódminősége és az alacsony szintű tesztlefedettség megfelelt a Termékkövetelményekben meghatározott elvárásoknak.

**Nem felelt meg**

A szoftvertermék kódminősége és az alacsony szintű tesztlefedettség nem felelt meg a Termékkövetelményekben meghatározott elvárásoknak.

#### Értékelési jelentés

Az értékelés eredményterméke a kódminőség és az alacsony szintű tesztlefedettség megfelelőségi jelentése (Értékelési jelentés).

Az Értékelési jelentés rögzíti, hogy az átadott vizsgálati jegyzőkönyv, illetve a hozzá csatolt vizsgálati eredmények megfelelnek-e a szoftvertermékkel szemben támasztott elvárásoknak.

#### ÁAFK Kr. hatálya alá eső fejlesztés

Abban az esetben, ha az IdomSoft technológiai csapata végez kódminőség vizsgálatot, akkor a minőségbiztosító annak az eredményét veszi át és értékeli.

Az értékelés elvégzéséhez szükséges a kódminőség vizsgálatról készült megfelelőségi szakmai vélemény.

#### Termékkövetelmények

| **Követelmény azonosítója** | **Követelmény megnevezése** | **Követelmény leírása** |
| --- | --- | --- |
| KM-T-01 | Sor lefedettség | A forráskódhoz készített tesztek kódra vetített lefedettsége almodulonként eléri a 60%-ot. |
| KM-T-02 | Karbantarthatósági besorolás | A karbantarthatóságot a gyanús kódrészek (code smells) száma, és azok kijavításának ideje határozza meg. A SonarQube alapértelmezett számítása szerint a szoftvertermék eléri legalább a C besorolást. |
| KM-T-03 | Megbízhatósági besorolás | A SonarQube beállításának megfelelően nincs blokkoló vagy kritikus besorolású programhiba. |
| KM-T-04 | Biztonsági besorolás | A SonarQube beállításának megfelelően nincs blokkoló vagy kritikus besorolású sérülékenységi hiba. |
| KM-T-05 | Duplikáció | A forráskódban előforduló duplikációk küszöbhatára nem haladja meg a 15%-ot. |

#### Módszertani leírás - KM-MOD-01:2022-1

#### Bevezetés

Jelen módszer célja, hogy új alkalmazásfejlesztés és teljes kódminőség vizsgálat esetén, a kódhoz készített tesztek futtatásra kerüljenek és a kódra vetített alacsony szintű teszt lefedettség és statikus kódminőség vizsgálat végrehajtása megtörténjen. A vizsgálat során feltárásra kerülnek a karbantarthatósági hiányosságok, programhibák és kódban szereplő sérülékenységek; amelyekről javítási javaslat megtalálható a jegyzőkönyv mellékletében.

#### Vizsgálati előfeltételek

A vizsgálati előfeltételeket a Nyújtott szolgáltatások bemeneti követelményeinek gyűjteménye tartalmazza, ahol jelen módszertan esetében a következő azonosítójú követelmények értelmezettek: KM-F-01, KM-F-02, KM-F-04, KM-F-05, KM-F-06, KM-F-07, KM-F-08, KM-F-09, KM-F-10.

#### A vizsgálati módszer során használt eszközök

* AlmaLinux (Arctic Sphynx)
* Gitea 4.2.1
* SonarQube 9.6
* SonarScanner 4.7

#### A vizsgálati módszer leírása

* A forráskód felépítésének megvizsgálása.
* A Git forráskód tároló szoftverben a projekt létrehozása, konfigurálása megfelelő beállításokkal.
* A leszállított forráskód elhelyezése a létrehozott, projekthez tartozó Git forráskód tároló projektben.
* A projekthez szükséges dedikált virtuális gép létrehozása, konfigurálása, majd a build végrehajtása, amennyiben szükséges.
* SonarQube kódminőség vizsgálatra alkalmas eszköz konfigurálása.
* SonarQube properties kiterjesztésű állomány megfelelő konfigurálása a fejlesztőkkel egyeztetettek szerint.
* A forráskódhoz készített tesztek futtatása a fejlesztők által meghatározott konfigurációs beállításokkal, melynek hatására előállnak a SonarQube számára értelmezhető jelentések.
* SonarScanner futtatása, ahol beolvasásra kerülnek a jelentések, és a szkennelésre kerül a forráskód.
* A forráskódban feltárt karbantarthatósági hiányosságok, programhibák, alacsony szintű teszt lefedettség, kódban szereplő sérülékenységek kiértékelése.
* A vizsgálat dokumentálása, jegyzőkönyv készítése.

#### Mérőeszközök

Az IdomSoft által kódminőség vizsgálatra használt statikus kódelemző szoftver a tesztek futtatására és a tesztek futtatási eredményének jelentés generálására nem alkalmas, viszont a kész jelentések megjelenítését lehetővé teszi. A komponens és komponens-integráció szintű tesztek futtatásához és jelentés eredmények generálásához szükséges eszköz programozási nyelvenként eltérő, így ezen eszközök használatához szakmai támogatás szükséges a fejlesztő szervezettől.

Amennyiben a fejlesztő szervezet alkalmazott a fejlesztés folyamatában lefedettség mérő eszközt, abban az esetben azon eszköznek a validáción és verifikáción kell átesni, azaz az eszköz lefedettség mérő módszerének igazolását és érvényesítését szükségszerűen kell elvégezni.

Abban az esetben, ha a fejlesztő szervezet a fejlesztési folyamatába nem építette be a kódhoz készített tesztek kódra vetített lefedettségének vizsgálatát, úgy az IdomSoft javaslatot tehet, mely eszköz beépítése valósuljon meg a fejlesztő szervezet által a kódminőség vizsgálathoz.

Amennyiben a fejlesztő szervezet nem alkalmazott alacsony szintű teszt lefedettség mérő eszközt, úgy lehetőséget biztosít az IdomSoft arra, hogy beépítésre kerüljön a projektbe a lefedettség mérő eszköz, elkerülve a leszállított forráskód felülírását, a program működésének megváltoztatását és a vizsgálat hamis eredmény kimenetelét. (Amennyiben az alkalmazott programnyelvben vagy keretrendszerben ez megvalósítható.)

#### Verifikálás és validálás

A vizsgálat elvégzéséhez a fejlesztő szervezet által alkalmazott alacsony szintű teszt lefedettség mérő eszköz igazolása (verifikálása) és érvényesítése (validálása) szükséges.

Az alkalmazott alacsony szintű teszt lefedettség mérő eszköz verifikálásának és validálásának folyamatát a Verifikációs és validációs kérelem benyújtásával szükséges elindítani.

#### Hibák feltárása

**Karbantarthatósági hiányosságok (Code smells) feltárása:**

Feltárásra kerülnek a karbantarthatóságot nagyban befolyásoló problémák, amelyek kihatással vannak a kód olvashatóságára és értelmezhetőségére is.

A SonarSource által fejlesztett és a minőségbiztosító által alkalmazott SonarQube kódminőség vizsgálatra alkalmas szoftver programnyelvenként értelmezett szabályait az adott alkalmazásverzió határozza meg. A kódminőség vizsgálat során a lehívás időpontjában érvényes alkalmazásverzió szabályrendszere érvényes. Továbbá a minőségbiztosító lefuttatja a kódminőség vizsgálatot az alkalmazott legújabb verzió szabályrendszerében érvényes szabályok figyelembevételével is. A vizsgálat során kritikus hibákon kívül minden, a verzióváltás hatására keletkezett új hibát a minőségbiztosító figyelmen kívül hagy. Kisebb stilisztikai, ellentmondásos vagy a kódminőséget szignifikánsan nem befolyásoló szabályt szintén figyelmen kívül hagy. Ezen szabályokat a minőségbiztosító valamennyi projekt esetében egyedileg bírálja felül.

**Programhibák (BUG) feltárása:**

Feltárásra kerülnek azok a kódolási hibák is, amelyek a program helyes működését befolyásolják. Kritikus esetben akár az szoftvertermék fő funkcióinak vagy a teljes szoftverterméknek a leállásához vezethetnek, így a váratlan hiba miatti leállás szinte minden esetben adatvesztéssel járhat. Kijavításuk erősen indokolt a nem várt program viselkedés elkerülése és stabil működés érdekében.

**Kódban szereplő biztonsági rések és sérülékenységek feltárása:**

A biztonsági rések és a sérülékenységek átfogó képet nyújtanak az szoftvertermék biztonságáról, vizsgálva az OWASP Top 10 2017, a CWE/SANS Top 25 sérülékenységeket:

* CWE: kiberbiztonsági sérülékenységek listája, amely hatóságilag kerül beszámozásra és megjelenítésre; metrikaként szolgál a biztonsági eszközök számára, és kiindulási alapként szolgál a gyengeségek azonosításához, enyhítéséhez és megelőzéséhez.
* OWASP: egy olyan non-profit szervezet, mely a szoftver biztonságát hivatott előmozdítani. Az ott fellelhető és eszközölt projektek révén valósítja meg, melyek témái széles spektrumon mozognak: audit módszertanok, fejlesztést támogató eszközök, követendő praktikák, szabvány javaslatok.

**Kódhoz készített tesztek kódra vetített lefedettségének vizsgálata:**

A forráskódhoz készített tesztek kódra vetített lefedettség küszöbhatára almodulonként 60%-os sor lefedettség, amennyiben ezen kritériumnak nem feleltethető meg a forráskód, abban az esetben a vizsgált rendszer nem teljesíti a követelményeket.

#### A vizsgálati módszer eredményterméke(i)

A vizsgálatok kimenete kizárólag a végrehajtása során keletkezett jegyzőkönyv.

**Vizsgálati jegyzőkönyv:**

A vizsgálat elvégzését követően egy jegyzőkönyv készül a vizsgálati jegyzőkönyv sablon alapján.

**Vizsgálati eredmények:**

A vizsgálati eredményeket a SonarQube program adatbázisából nyerjük ki egy saját fejlesztésű eszközzel. Az eszköz képes a SonarQube projektjei közül választani és .xlsx formátumban riportot generálni adott projekt vizsgálati eredményeiből. Az említett .xlsx fájl csatolásra kerül a vizsgálati jegyzőkönyvhöz. A táblázat tartalma:

* 1. munkalap: Általános
  + component - a vizsgált komponens elérési útja a projekten belül
  + name – a vizsgált komponens neve
  + coverage – adott komponensre számított sorlefedettség
  + duplication – adott komponensre számított duplikációs ráta
* 2. munkalap: Bug (programhibák)
  + rule – a Sonar szabály azonosítója
  + message – a Sonar szabály leírása
  + hint – a Sonar szabály KDIV által írt magyarázata
  + type – a hiba típusa
  + severity – a hiba súlyossága
  + component – a komponens elérési útja
  + line – a hiba helye a komponensen belül
  + effort – a hiba kijavítására kalkulált időtartam
  + status – a hiba státusza
* 3. munkalap: Vulnerability (sérülékenységi hibák)
  + rule – a Sonar szabály azonosítója
  + message – a Sonar szabály leírása
  + hint – a Sonar szabály KDIV által írt magyarázata
  + type – a hiba típusa
  + severity – a hiba súlyossága
  + component – a komponens elérési útja
  + line – a hiba helye a komponensen belül
  + effort – a hiba kijavítására kalkulált időtartam
  + status – a hiba státusza
* 4. munkalap: Code Smell (karbantarthatósági hibák)
  + rule – a Sonar szabály azonosítója
  + message – a Sonar szabály leírása
  + hint – a Sonar szabály KDIV által írt magyarázata
  + type – a hiba típusa
  + severity – a hiba súlyossága
  + component – a komponens elérési útja
  + line – a hiba helye a komponensen belül
  + effort – a hiba kijavítására kalkulált időtartam
  + status – a hiba státusza
* 5. munkalap: Security Hotspot (biztonsági hibák)
  + rule – a Sonar szabály azonosítója
  + message – a Sonar szabály leírása
  + hint – a Sonar szabály KDIV által írt magyarázata
  + type – a hiba típusa
  + severity – a hiba súlyossága
  + component – a komponens elérési útja
  + line – a hiba helye a komponensen belül
  + effort – a hiba kijavítására kalkulált időtartam
  + status – a hiba státusza

#### Módszertani leírás - KM-MOD-02:2022-1

#### Bevezetés

Jelen módszer célja, hogy alkalmazás továbbfejlesztés és rész-kódminőség vizsgálat esetén, a kódhoz készített tesztek futtatásra kerüljenek és a kódra vetített alacsony szintű teszt lefedettség és statikus kódminőség vizsgálat végrehajtása megtörténjen. A vizsgálat során feltárásra kerülnek a karbantarthatósági hiányosságok, programhibák és kódban szereplő sérülékenységek; amelyekről javítási javaslat megtalálható a jegyzőkönyv mellékletében.

#### Vizsgálati előfeltételek

A vizsgálat előfeltételeket a Nyújtott szolgáltatások bemeneti követelményeinek gyűjteménye tartalmazza, ahol jelen módszertannál a következő azonosítójú követelmények értelmezettek: KM-F-01, KM-F-03, KM-F-04, KM-F-05, KM-F-06, KM-F-07, KM-F-08, KM-F-09, KM-F-10.

#### A vizsgálati módszer során használt eszközök

* AlmaLinux (Arctic Sphynx)
* Gitea 4.2.1
* SonarQube 9.6
* SonarScanner 4.7

#### A vizsgálati módszer leírása

* A forráskód felépítésének megvizsgálása.
* A Git forráskód tároló szoftverben a projekt létrehozása, konfigurálása megfelelő beállításokkal.
* A leszállított forráskód elhelyezése a létrehozott, projekthez tartozó Git forráskód tároló projektben. Ahol külön branchbe kerül az alkalmazás továbbfejlesztése során előállt forráskód és a fejlesztés előtti forráskód.
* A projekthez szükséges dedikált virtuális gép létrehozása, konfigurálása, majd a build végrehajtása, amennyiben szükséges.
* SonarQube kódminőség vizsgálatra alkalmas eszköz konfigurálása.
* SonarQube properties kiterjesztésű állomány megfelelő konfigurálása a fejlesztőkkel egyeztetettek szerint.
* A forráskódhoz készített tesztek futtatása branchenként, a fejlesztők által meghatározott konfigurációs beállításokkal, melynek hatására előállnak a SonarQube számára értelmezhető jelentések mind a továbbfejlesztés során keletkezett kódra, mind a továbbfejlesztés előtti kódra.
* SonarScanner futtatása, ahol beolvasásra kerülnek a jelentések, és a szkennelésre kerül a forráskód.
* Az alkalmazás különbséget tud tenni a két kódbázis között, a vizsgálatot csak a továbbfejlesztés során keletkezett kódra értelmezzük.
* A forráskódban feltárt karbantarthatósági hiányosságok, programhibák, alacsony szintű teszt lefedettség, kódban szereplő sérülékenységek kiértékelése.
* A vizsgálat dokumentálása, jegyzőkönyv készítése.

#### Mérőeszközök

Az IdomSoft által kódminőség vizsgálatra használt statikus kódelemző szoftver a tesztek futtatására és a tesztek futtatási eredményének jelentés generálására nem alkalmas, viszont a kész jelentések megjelenítését lehetővé teszi. A komponens és komponens-integráció szintű tesztek futtatásához és jelentés eredmények generálásához szükséges eszköz programozási nyelvenként eltérő, így ezen eszközök használatához szakmai támogatás szükséges a fejlesztő szervezettől.

Amennyiben a fejlesztő szervezet alkalmazott a fejlesztés folyamatában lefedettség mérő eszközt, abban az esetben azon eszköznek a validáción és verifikáción kell átesni, azaz az eszköz lefedettség mérő módszerének igazolását és érvényesítését szükségszerűen kell elvégezni.

Abban az esetben, ha a fejlesztő szervezet a fejlesztési folyamatába nem építette be a kódhoz készített tesztek kódra vetített lefedettségének vizsgálatát, úgy az IdomSoft javaslatot tehet, mely eszköz beépítése valósuljon meg a fejlesztő szervezet által a kódminőség vizsgálathoz.

Amennyiben a fejlesztő szervezet nem alkalmazott alacsony szintű teszt lefedettség mérő eszközt, úgy lehetőséget biztosít a IdomSoft arra, hogy beépítésre kerüljön a projektbe a lefedettség mérő eszköz, elkerülve a leszállított forráskód felülírását, a program működésének megváltoztatását és a vizsgálat hamis eredmény kimenetelét. (Amennyiben az alkalmazott programnyelvben vagy keretrendszerben ez megvalósítható.)

#### Verifikálás és validálás

A vizsgálat elvégzéséhez a fejlesztő szervezet által alkalmazott alacsony szintű teszt lefedettség mérő eszköz igazolása (verifikálása) és érvényesítése (validálása) szükséges.

Az alkalmazott alacsony szintű teszt lefedettség mérő eszköz verifikálásának és validálásának folyamatát a Verifikációs és validációs kérelem benyújtásával szükséges elindítani.

#### Hibák feltárása

**Karbantarthatósági hiányosságok (Code smells) feltárása:**

Feltárásra kerülnek a karbantarthatóságot nagyban befolyásoló problémák, amelyek kihatással vannak a kód olvashatóságára és értelmezhetőségére is.

A SonarSource által fejlesztett és a minőségbiztosító által alkalmazott SonarQube kódminőség vizsgálatra alkalmas szoftver programnyelvenként értelmezett szabályait az adott alkalmazásverzió határozza meg. A kódminőség vizsgálat során a lehívás időpontjában érvényes alkalmazásverzió szabályrendszere érvényes. Továbbá a minőségbiztosító lefuttatja a kódminőség vizsgálatot az alkalmazott legújabb verzió szabályrendszerében érvényes szabályok figyelembevételével is. A vizsgálat során kritikus hibákon kívül minden, a verzióváltás hatására keletkezett új hibát a minőségbiztosító figyelmen kívül hagy. Kisebb stilisztikai, ellentmondásos vagy a kódminőséget szignifikánsan nem befolyásoló szabályt szintén figyelmen kívül hagy. Ezen szabályokat a minőségbiztosító valamennyi projekt esetében egyedileg bírálja felül.

**Programhibák (BUG) feltárása:**

Feltárásra kerülnek azok a kódolási hibák is, amelyek a program helyes működését befolyásolják. Kritikus esetben akár az szoftvertermék fő funkcióinak vagy a teljes szoftverterméknek a leállásához vezethetnek, így a váratlan hiba miatti leállás szinte minden esetben adatvesztéssel járhat. Kijavításuk erősen indokolt a nem várt program viselkedés elkerülése és stabil működés érdekében.

**Kódban szereplő biztonsági rések és sérülékenységek feltárása:**

A biztonsági rések és a sérülékenységek átfogó képet nyújtanak az szoftvertermék biztonságáról, vizsgálva az OWASP Top 10 2017, a CWE/SANS Top 25 sérülékenységeket:

* CWE: kiberbiztonsági sérülékenységek listája, amely hatóságilag kerül beszámozásra és megjelenítésre; metrikaként szolgál a biztonsági eszközök számára, és kiindulási alapként szolgál a gyengeségek azonosításához, enyhítéséhez és megelőzéséhez.
* OWASP: egy olyan non-profit szervezet, mely a szoftver biztonságát hivatott előmozdítani. Az ott fellelhető és eszközölt projektek révén valósítja meg, melyek témái széles spektrumon mozognak: audit módszertanok, fejlesztést támogató eszközök, követendő praktikák, szabvány javaslatok.

**Kódhoz készített tesztek kódra vetített lefedettségének vizsgálata:**

A forráskódhoz készített tesztek kódra vetített lefedettség küszöbhatára almodulonként 60%-os sor lefedettség, amennyiben ezen kritériumnak nem feleltethető meg a forráskód, abban az esetben a vizsgált rendszer nem teljesíti a követelményeket.

#### A vizsgálati módszer eredményterméke(i)

A vizsgálatok kimenete kizárólag a végrehajtása során keletkezett jegyzőkönyv.

**Vizsgálati jegyzőkönyv:**

A vizsgálat elvégzését követően egy jegyzőkönyv készül a vizsgálati jegyzőkönyv sablon alapján.

**Vizsgálati eredmények:**

A vizsgálati eredményeket a SonarQube program adatbázisából nyerjük ki egy saját fejlesztésű eszközzel. Az eszköz képes a SonarQube projektjei közül választani és .xlsx formátumban riportot generálni adott projekt vizsgálati eredményeiből. Az említett .xlsx fájl csatolásra kerül a vizsgálati jegyzőkönyvhöz. A vizsgálati eredménytermékek esetében fontos ügyelni arra, hogy csak a különbségképzéssel meghatározott fájlok halmazára futtatott vizsgálat eredményeit tekintsük vizsgálati eredménynek.

A táblázat tartalma:

* 1. munkalap: Általános
  + component - a vizsgált komponens elérési útja a projekten belül
  + name – a vizsgált komponens neve
  + coverage – adott komponensre számított sorlefedettség
  + duplication – adott komponensre számított duplikációs ráta
* 2. munkalap: Bug (programhibák)
  + rule – a Sonar szabály azonosítója
  + message – a Sonar szabály leírása
  + hint – a Sonar szabály KDIV által írt magyarázata
  + type – a hiba típusa
  + severity – a hiba súlyossága
  + component – a komponens elérési útja
  + line – a hiba helye a komponensen belül
  + effort – a hiba kijavítására kalkulált időtartam
  + status – a hiba státusza
* 3. munkalap: Vulnerability (sérülékenységi hibák)
  + rule – a Sonar szabály azonosítója
  + message – a Sonar szabály leírása
  + hint – a Sonar szabály KDIV által írt magyarázata
  + type – a hiba típusa
  + severity – a hiba súlyossága
  + component – a komponens elérési útja
  + line – a hiba helye a komponensen belül
  + effort – a hiba kijavítására kalkulált időtartam
  + status – a hiba státusza
* 4. munkalap: Code Smell (karbantarthatósági hibák)
  + rule – a Sonar szabály azonosítója
  + message – a Sonar szabály leírása
  + hint – a Sonar szabály KDIV által írt magyarázata
  + type – a hiba típusa
  + severity – a hiba súlyossága
  + component – a komponens elérési útja
  + line – a hiba helye a komponensen belül
  + effort – a hiba kijavítására kalkulált időtartam
  + status – a hiba státusza
* 5. munkalap: Security Hotspot (biztonsági hibák)
  + rule – a Sonar szabály azonosítója
  + message – a Sonar szabály leírása
  + hint – a Sonar szabály KDIV által írt magyarázata
  + type – a hiba típusa
  + severity – a hiba súlyossága
  + component – a komponens elérési útja
  + line – a hiba helye a komponensen belül
  + effort – a hiba kijavítására kalkulált időtartam
  + status – a hiba státusza

#### Módszertani leírás - KM-MOD-03:2022-1

#### Bevezetés

Jelen módszer célja, a Megrendelő Szoftvertermékéhez a külső fél által készített kódminőség és alacsony szintű teszt lefedettség vizsgálat átvizsgálása.

#### Vizsgálati előfeltételek

A vizsgálat előfeltételeket a Nyújtott szolgáltatások bemeneti követelményeinek gyűjteménye tartalmazza, ahol jelen módszertan esetében a következő azonosítójú követelmények értelmezettek: KM-F-11, KM-F-12.

#### A vizsgálati módszer során használt eszközök

Az elkészült dokumentáció megnyitására és kezelésére alkalmas irodai szoftver (MS Word, MS Excel, PDF reader, MS PPT vagy egyéb irodai szoftver).

#### A vizsgálati módszer leírása

Külső fél által a Szoftverterméken végrehajtott kódminőség és alacsony szintű teszt lefedettség vizsgálat eredményeinek értékelése történik. A vizsgálatot végző kolléga megbizonyosodik arról, hogy az átadott eredmények általa már korábban verifikált és validált eszközökkel és módszerekkel álltak elő.

Az átvizsgálás elvégzéséhez a külső fél által alkalmazott eszközt és módszert verifikálni és validálni kell, tehát az alkalmazott kódminőség és alacsony szintű teszt lefedettség vizsgálati eszköz és módszer igazolása és érvényesítése szükséges. Amennyiben az alkalmazott eszköz és módszer még nem rendelkezik az IdomSoft által kiállított verifikációs és validációs nyilatkozattal, úgy a verifikációs és validációs eljárás lefolytatását szükséges kérelmezni.

Az alkalmazott kódminőség és alacsony szintű teszt lefedettség vizsgálati eszköz és módszer verifikálásának és validálásának folyamatát a Verifikációs és validációs kérelem benyújtásával szükséges elindítani.

#### A vizsgálati módszer eredményterméke(i)

Feljegyzés átvizsgálásról:

A külső fél által végzett kódminőség és alacsony szintű teszt lefedettség vizsgálat eredményeit a vizsgálatot értékelő kolléga átvizsgálja és feljegyzést készít az átvizsgálásról, mely tartalmazza a minőségbiztosító megállapítását az átadott eredmények elfogadásáról vagy elutasításáról.