# Cím Szoftver tesztelés, szoftver tanúsítás mint szolgáltatás a számítási felhőben

Kepzesi\_szint: MSc

Temavezeto: Molnár Bálint

Email: molnarba@inf.elte.hu

Temakor: E-government,Rendszerfejlesztés,rendszermodellezés

A "TaaS- Testing as a Service" - tesztelés mint szolgáltatás már létező

szolgáltatási módozat a számítási felhőből. A szoftverek

bevizsgálásának, tanűsításának leehtséges megoldásai, nyílt forráskódú

eszköz készletek, leehtséges üzleti modellek feldolgozása, elemzése a

tárgya e témának.

Lhetséges feldolgozási módok:

Nyílt forráskódú rendszerekből pilot / prototípus összeéptésének terve -

Pl. O-O, UML módszertani alapokon.

Szervezeti és Szoftver Architektúraák leírása a lehetséges megoldásokról.

UML Deployment (telepítési diagram), TOGAF

Lehetséges szervezeti /üzeleti folyamatok megtervezése (Business Process

engineering), leírása módszertani alapokon pl. BPMN 2.0, ARIS EPC, UML

Activity diagram

A funkcionális és nem funkcionáli sszempontok formális és félig-formális

módszereinek áttekintése, elemzése, egyes részterületekre valamilyen

módszer kialakítása

1. Felhő és vállalkozás támogatás

A számítási felhőből, felhőben elhelyezendő üzleti szolgáltatások és

számítási felhő „infrastruktrúális szolgáltatásokra teszt környezet

kialakítása, amelyet a felhőből lehetne nyújtani (TaaS, Test as a

Service).

1.2.1. Tesztelési szolgáltatás a számítási felhőből

A piaci szereplők, szoftver gyártók támogatására, egy olyan számítási

felhőből nyújtott üzleti szolgáltatás kialakítása, amelynek a révén a

potenciális szállítók, gyártók saját maguk tesztelhetik, készíthetik elő

terméküket a szoftver minőségtanúsításra, az adott számítási felhőbe

történő elhelyezésére, befogadására (adoptálására).

A partnerek a szolgáltatást valójában „béta” tesztelésre, az ISO 9126

(pl. http://sqi.hu/hu/our-sqm-approaches/iso-9126/ ) szabványban

megfogalmazott nem funkcionális tulajdonságok vizsgálatára, de a különböző

egyéb szabványok, előírások, szabályozásoknak történő megfelelés

vizsgálatára is igénybe vehetik a szolgáltatást. Azaz a szélesebb

értelemben vett „megfelelőség” („compliance”), szavatosság garantálására,

biztosíték nyújtására („assurance”), projekt folyamat audit, projekt

termék előállítás auditra, a szoftver fejlesztési folyamat érettségére

vonatkozó vizsgálatok elvégezhetőségét is támogatnia kell e környezetnek.

2. Formális és félig formális architektúra és rendszermodellezési

eljárások és modellek kialakítása

A számítási felhő szolgáltatási modelljeinek, a vállalkozások szolgáltatás-

központú működésének informatika módszertani, modellezési eljárásainak

kialakítása, amelyekkel meta-modellek, referencia modellek készíthetők.

A modellek nem a formális módszerekkel történő (program helyesség

bizonyítás jellegű) vizsgálatokat megtámogatását célozzák meg, hanem a

szervezeti/üzleti folyamatoktól a technológiai, logikai/alkalmazási

platformokig nyúló tipikus megoldási és modell mintázatok feltárását annak

érdekében, hogy automatizált és félig automatizált módszerekkel a tipikus

hibák, konstrukciók feltárhatók legyenek.

Az architektúra modellek:

— Zachman;

— TOGAF;

— SOA, RESTful, Web szolgáltatás;

— Számítási felhő/számítási magánfelhő

— Formális szoftver architektúra modellek.

— Programozási nyelvek, paradigmák architektúra mintázatai (GoF,

(Gang of Four), GRASP, stb.

Rendszer modellek:

— szervezeti (vállalati, üzleti) folyamat modellek (BPMN 2.0, BPEL

stb.);

— UML/OOA-OOD modellek;

— Programozási nyelvek, paradigmák architektúra mintázatai.

A számítási felhő szolgáltatatásai egyrészt egy üzleti/vállalati/

vállalkozási igényt elégítenek szolgáltatás-központú megközelítésben,

másrészt az igényelt szolgáltatásokat a számítási felhő szolgáltatási

palettájának valamelyik formája segítségével nyújtják.

Egy új, egy adott magánfelhőbe beillesztendő szolgáltatás szoftver

megvalósítása az előbbiek fényében lehet egy számítási felhő architektúra

szolgáltatás, másrészt egy üzleti szolgáltatás társadalmi, gazdasági

entitások, vállalkozások számára, amely a számítási felhő architektúra

alapszolgáltatásaira támaszkodik.

Mindkét esetben komoly kompatibilitási, interoperabilitási, illeszkedési,

együttműködési, és teljesítmény kérdések merülnek fel egy magánfelhőbe

történő, szoftverre leképezett szolgáltatás bevezetése és beillesztése

kapcsán. Vajon a magánfelhő technológiai architektúrája be tudja-e fogadni

az esetlegesen új alap számítási felhő szolgáltatást. Szervezeti

(vállalati, üzleti) szolgáltatás esetében a magánfelhő milyen virtualizált

környezetébe, milyen megoldással lehet elhelyezni a szolgáltatást.

2.2. A informatikai, műszaki háttér

A számítási felhőn keresztül nyújtott szolgáltatások valójában két

dimenzióban fogalmazhatók meg. Az egyik dimenzió a felhő

számítástechnikai, vagy felhő infrastruktúraszolgáltatások. Ezeket a nagy

gyártok, marketingesei illetve értékesítési szakemberei három fő

kategóriára szűkítik le. Nevezetesen:

— SaaS, a szoftver mint szolgáltatás,

— PaaS, a platform mint szolgáltatás;

— IaaS, az infrastruktúra mint szolgáltatás (vagyis hardver és

kiszolgáló gépek stb.).

 A felhő számítástechnikai szolgáltatások ennél sokkal finomabban

bonthatók. A felhő-számítástechnikában tulajdonképpen mindent

szolgáltatásnak lehet tekinteni (XaaS) :

1. Tesztelés mint szolgáltatás (TaaS, Testing-as-a-service);

2. Szoftver mint szolgáltatás (SaaS , Software as a Service)

3. Platform mint szolgáltatás (PaaS, Platform-as-a-service);

4. Hardver mint szolgáltatás (HaaS, Hardver-as-a-service);

5. Infrastruktúra mint szolgáltatás (IaaS,

Infrastructure-as-a-service);

6. Alkalmazás mint szolgáltatás (ApaaS, Application-as-a-service);

7. Köztesrendszer mint szolgáltatás (MiaaS, Middleware-as

a-service);

8. Mashup mint szolgáltatás (MaaS, Mashup-as-a-service);

9. Biztonság mint szolgáltatás (Security-as-a-service);

10. Együttműködés mint szolgáltatás (CaaS,

Collaboration-as-a-service);

11. Információ mint szolgáltatás (InfoaaS,

Information-as-a-service);

12. Keretrendszer mint szolgáltatás (FaaS, Framework-as-a-service);

13. Modellezés és Meta-modellezés mint szolgáltatás (MMaaS,

Modeling & Metamodeling-as-a-service).

14. Adattároló mint szolgáltatás (StaaS, Storage-as-a-service);

15. Személyazonosítás mint szolgáltatás (IPMaaS, Identity & Policy

Management-as-a-Service);

16. Vállalat/Szervezet irányítás mint szolgáltatás (EaaS,

Enterprise-as-a-Service);

17. Üzleti tevékenységek mint szolgáltatás (BaaS,

Business-as-a-service);

18. Asztali gép mint szolgáltatás (DaaS, Destop-as-a-service);

19. Adatbázis mint szolgáltatás (DBaaS, Database-as-a-service);

20. Folyamat mint szolgáltatás (PraaS, Process-as-a-service);

21. Integrálás mint szolgáltatás (Integration-as-a-service);

22. Vezetés / irányítás mint szolgáltatás

(Management/governance-as-a-service);

A fentebbi (XaaS) szolgáltatások tehát a számítási felhő

„infrastrukturális” szolgáltatásainak tekinthetők.

A számítási felhő társadalmi és gazdasági entitások felé nyújtott

szolgáltatásai a szolgáltatatás központú szoftver architektúra felfogásban

értelmezhetők. Ezt SzOA-val szokták jelölni, de a mai korszerű SzOA

megközelítés értelmében érdemes felfogni a szervezeti architektúra üzleti/

szervezeti folyamataitól a technológia architektúráig, ami a számítási

felhő környezetben a felhő architekturális építőelemeit jelenti. A

számítási felhő (infrastrukturális) szolgáltatásai és a számítási felhőben

elhelyezett szervezeti (vállalati, üzleti) szolgáltatások egymásra

ortogonálisok.

Nem-funkcionális minőségi tulajdonságok

A tesztelendő szolgáltatás valamilyen architektúra paradigmát és

mintázatokat alkalmaz. A szoftver minőség tesztelési szolgáltatás egyik

feladata, hogy lehetővé tegye olyan vizsgálatok elvégzését, amellyel

egyrészt az ISO 9126 szerinti nem funkcionális tulajdonságok, és az

alkalmazott architektúra építőelemek megfelelőségére vonatkozó elemzéseket

el lehessen végezni:

Fogalom Meghatározás

Hatékonyság A rendszer vagy egy alkotórésze által nyújtott teljesítmény és

a felhasznált erőforrások viszonya. A hatékonyság a rendszer idő-

dimenzióban nyújtott viselkedéséhez kötődik tipikusan, nevezetesen a

válaszidő, adatfeldolgozási idő, rendszer (feladat, munka)

áteresztőképessége, továbbá az erőforrások felhasználásának mikéntjét

jelenti, azaz a felhasznált erőforrások volumene, az igénybevétel ideje.

Funkcionalitás A rendszer funkcionalitását a rendszer funkcióinak halmaza

és a specifikált tulajdonságaik jellemzik. A funkciók azok, amelyek

megfelelnek a szervezet működése olyan igényeinek, amelyeket kifejezetten

közöltek, illetve amelyek következtek vagy levezethetőek voltak az

előbbiekből.

 További idevágó jellemzők:

 · alkalmasság

 · pontosság, szabatosság

 · együttműködési képesség

 · igényekhez való illeszkedés;

 · biztonság;

 · nyomon követhetőség.

Biztonság A rendszer azon képessége, hogy megakadályozza az adatokhoz és a

programokhoz való jogosulatlan hozzáféréseket, akár szándékos legyen akár

véletlen.

Megbízhatóság A rendszer azon képessége, hogy egy bizonyos idő

intervallumra vonatkozóan a rendszer folyamatosan tudja szolgáltatni az

előírt teljesítményt a meghatározott feltételek mellett.

 A meghibásodások között eltelt átlagos idővel szokták mérni.

 További idevágó jellemzők:

 · a rendszer hibatűrő képessége

 · visszaállíthatóság;

 · rendelkezésre állás;

 · szolgáltatáscsökkentésre való készség,

 · a rendszer kiérleltsége.

Karbantartha–tóság A rendszer egészének vagy egyes részeinek módosításához

szükséges erőfeszítések nagysága jellemzi. A rendszer–módosítások közé

értendők a hibajavítás jellegű, korrekciós karbantartások, a tovább

fejlesztés vagy a környezet változása miatti adaptációs fejlesztés,

továbbá a követelmény és funkcionális specifikációban végzendő javítások.

 További idevágó jellemzők:

 · elemezhetőség egyszerűsége;

 · változtathatóság, módosíthatóság mértéke;

 · a rendszer stabilitása;

 · tesztelhetősége;

 · kézben tarthatósága (üzemeltetés, működés közben);

 · újra felhasználhatósága.

Hordozhatóság A rendszer azon sajátossága, hogy milyen könnyen lehet a

rendszer egészét vagy bizonyos részeit egy teljesen más környezetbe

(technológiai és szervezeti értelemben is) átültetni.

 További idevágó jellemzők:

 · adaptációs képesség;

 · üzembe-helyezés egyszerűsége;

 · kicserélhetősége, helyettesíthetősége.

Használhatóság A leendő felhasználóknak, a rendszer aktorainak, mekkora

nehézséget okoz a rendszer használata, mennyire egyszerű, könnyű a

rendszerrel dolgozni.

 További idevágó jellemzők:

 · megérthetőség;

 · megtanulhatóság;

 · üzemeltethetőség, működtetés egyszerűsége;

 · a rendszer működésének átláthatósága, nyíltsága;

 · testre-szabhatósága;

 · megjelenésének vonzereje;

 · a működés átláthatósága;

 · segítség és támogatás nyújtási szolgáltatásainak színvonala;

 · felhasználó-barát-e.

(Megj.: Ezeknek további részletezése az ISO 9126 számú szabványban

található meg.)

2.3.1.1. A tesztelendő rendszer architektúra nézetei

Az egyes architektúra területeket részletesebben fel lehet bontani,

illetve az igényeknek megfelelő nézetek alakíthatók ki.

1. Az üzleti (szervezeti) architektúra nézet a végfelhasználó igényeinek,

szempontjainak megfogalmazását célozza.

2. A biztonsági architektúra nézet a rendszer biztonsági architekturális

kérdéseivel foglakozik.

3. A szoftvertervező architektúra nézet az új szoftver rendszerek

fejlesztéséhez szükséges ábrázolást jelenti.

4. A rendszermérnöki architektúra nézet a szoftver és hardver komponensek

működő rendszerré történő összeépítéséhez szükséges leírást jelenti.

5. A hálózattervezői és kommunikáció architektúra nézet a hálózati,

kommunikációs, távközlési elemek olyan leírását jelenti, amely

leegyszerűsíti a hálózati, kommunikációs elemekre a továbbfejlesztési

tervek és a műszaki tervek elkészítését.

6. Az adatáramlási architektúra nézet az adatok tárolásával,

visszakeresésével, feldolgozásával, archiválásával és biztonsági kérdéseit

írja le.

7. A vállalkozás (szervezet) irányítási architektúra nézet a működéssel

(„operations”), adminisztrációval és a rendszer igazgatásával foglalkozik.

8. A beszerzési architektúra nézet a „polcról levehető”, dobozos szoftver

és hardver termékek beszerzésével kapcsolatos architekturális kérdéseket

ábrázolja.

2.3.1.2. Tesztelendő rendszer funkcionális tulajdonságok

Minőségi/tanúsítási kritériumok

A minőségi/átvételi kritériumokat az alkalmazott módszertan és/vagy

szabvány a termékekre rögzíti, előírja.

A konkrét projektre meghatározott minőségi elvárások, követelmények

ismeretében kell a minőségi/tanúsítási kritériumokat a minőségbiztosítási

terv elkészítésekor a projektben előállítandónak megnevezett „termékre

szabni”.

Mind a minőségi elvárások, követelmények, mind az tanúsítási kritériumok

teljesülése ellenőrizendő.

A szolgáltatás szoftver megvalósítása során előállítandó modellek,

termékek bevizsgálására is támogatási környezetet kell kialakítani. A

strukturált és az UML/ O-O alapú módszertanok legfontosabb termékei,

amelyekre a bevizsgálást támogató módszereket kell kialakítani.

1. Szervezeti (vállalati, üzleti), ügyviteli folyamatmodell

2. Vízió (RUP) / Rendszerszervezési javaslat (hagyományos)

3. Használati eset modell (RUP) / Követelményspecifikáció (hagyományos)

4. Magas szintű elemzési modell

5. Részletes elemzési modell

6. Használati esetek megvalósításának tervei

7. Felhasználói felület tervei

8. Logikai rendszerterv

9. Fizikai rendszerterv

10. Prototípus

11. Felhasználói kézikönyv (elektronikus, súgó)

12. Üzemeltetési kézikönyv (elektronikus, súgó)

2.3.2. A tesztelési rendszer környezet kialakítása

Az korábban összefoglalt szempontrendszer alapján a tanúsításhoz

szükséges komplex szoftver környezetek előállítása, előrekonfigurált

állapotban, virtuális gépben elhelyezhető módon.

A bevizsgálási eljárás a szoftver tulajdonságok széles körére vonatkozik,

a funkcionális sajátosságoktól kezdődően a nem-funkcionális

jellegzetességek elemzéséig.

A szabványoknak, a szabványok műszaki leírásainak, a szabványok

kiegészítéseinek, az iparban elfogadott ellenőrzési és vizsgálati

útmutatóknak és szabályozásoknak történő megfelelés vizsgálata, a

benyújtott termék auditjának nevezhető. A környezetnek e tevékenységeket

is támogatni kell. Az elsősorban szóba jövő szabályozások a következők:

1. COBIT 4 és 5 változat;

2. ISO 27000 biztonságirányítási szabvány család;

3. A közös szempontrendszer (Common Criteria for Information Technology

Security Evaluation (abbreviated as Common Criteria or CC) is an

international standard (ISO/IEC 15408)).

4. Dokumentációs szabványok (ISO 6592);

5. Szoftver fejlesztés szabványok

a. A COBIT és az ITIL is érinti a szoftver beszerzés folyamatait akár

fejlesztés, akár valamilyen vásárlási formáról is legyen szó. A szoftver

fejlesztésre vonatkozó szabvány a 12207, amely viszonylag részletesen de

nem egyetlen módszertant előnybe részesítve szabályozza a szoftver

fejlesztési folyamatot. A COBIT és ITIL folyamatoknál sokkal

részletesebben bontja ki a fejlesztési lépéseket, ezért a vonatkozó COBIT

és ITIL folyamatoknál kell a 12207 részletesebb szabályozását.

6. ISO 12207 a szoftver fejlesztésekre, beszerzésekre, a külső szállítókra

is vonatkoztatva

7. A bevezetett informatikai folyamatok folyamatos javításának

megteremtése (ISO 15504)

8. ISO 20000 ITIL-szabvány

9. ISO 9126 (pl. http://sqi.hu/hu/our-sqm-approaches/iso-9126/ )

A szabályozásnak, szabványoknak történő megfelelőség vizsgálatot is

alkalmaz eszköz készlettel támogatni kell, amelyek az egyszerű irodai

szoftver alkalmazásoktól a „lényegi tesztek” megfogalmazásáig („material

testing”) , előállításáig és félig automatikus végrahjtásáig terjed.

A komplex tesztelési környezetnek a bevizsgálási feladatokat kell

támogatnia. A feladatok nagy részére automatizált eszközök találhatók,

gyakran nyílt forráskódú megvalósítások is. Az alkalmazott kutatási

feladat ebben az esetben a következő:

1. A rendelkezésre álló eszközök adaptálása, testre szabása az új

technológiai környezetre és az új „Szoftver tesztelés szolgáltatás”

kontextusára.

2. A formális, félig-formális modellek, architektúra megközelítések

alapján feltárt olyan tulajdonságok bevizsgálásra alkalmas teszt eszközök

kifejlesztése, amelyekre nem találhatók olyan tesztelési alkalmazások,

amelyek sikeresen testre szabhatók volnának.