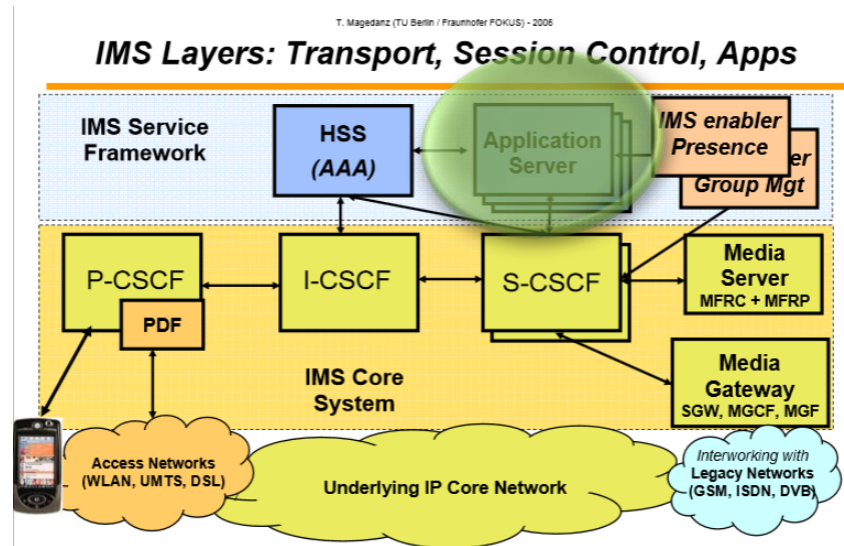


V&V @ Telecommunication – A Nokia telefon- applikációs szerver (OpenTAS) élelciklusának bemutatása

- Tündik Máté Ákos
- 03-12-2015

VoLTE / LTE , IMS, TAS

- IP Multimedia Subsystem (IP Multimédia alhálózat):
 - ❑ csomagkapcsolt hálózatok fölé kialakított fedőhálózat
 - ❑ Különböző szabványok közötti átjárást biztosít, így egyaránt kiépíthető bármilyen vezetékes (DSL, kábel), fix-vezeték nélküli (WLAN) illetve mobilhálózatok (GSM, UTMS) integrálásával.
 - ❑ Bármilyen készülékkel csatlakozhat rá a felhasználó: mobiltelefon, PDA, PC, tablet
 - ❑ A felhasználó akár az összes telefonhívását IP-hálózaton keresztül lebonyolíthatja ->költségsökkentés
 - ❑ Alkalmazott jelzésprotokollok: pl. SIP, SDP, Diameter
- Az alkalmazás szerver felel a végfelhasználók felé nyújtott (multimédiás) szolgáltatásokért és alkalmazásokért.
- LTE (4G) – t először adatkapcsolatok megvalósítására használták Gyorsabb letöltés, sok felhasználót tud egyidejűleg kiszolgálni.
→ VoLTE: Cél, hogy hanghíváshoz a 2G/3G hálózatok helyett használjuk az IP alapú hangátvitelt. Gyorsabb a hívásfelépítés, jobb a hangminőség (HD Voice)



Forrás: http://www.tilb.sze.hu/tilb/targyak/NGM_TA011_1/FOKUS-IMS-Tutorial-public.pdf

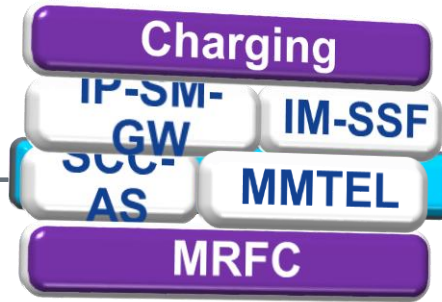
TAS funkcionalitások

SMS szolgáltatás LTE felhasználók részére (IP/LTE)

- Üzenetkézbesítés
- Számlázás

VoLTE szolgáltatás-folytonosságot biztosító és routing funkciók

- Handover VoLTE és CS hálózat között hívás közben (SR-VCC)
- Döntést hoz arról, hogy IMS hálózatba irányítódjon a hívás, vagy sem (Homing)
- Döntést hoz arról, hogy a hívás LTE vagy CS hálózaton menjen végbe (T-ADS)



OpenTAS



Együttműködés meglévő Intelligens szolgáltatásokat biztosító SCP-kkel

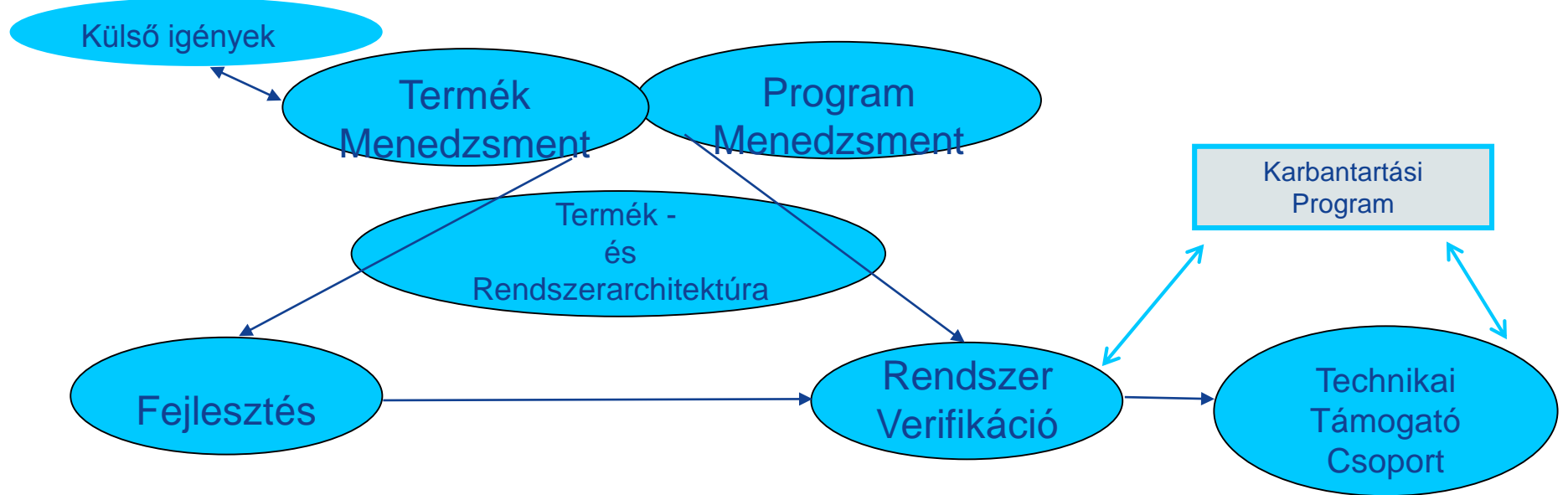
- IM-SSF (3GPP)
- INAP/CAP interfész támogatás

MMTEL (audio/video) szolgáltatások:

- Hívásfelépítés/-bontás
- Hívásátirányítás
- Hívások szűrése
- announcement-ek / tone-ok bejátszása (MGW/MRF)

*The Service Centralisation & Continuity Application Server (SCC-AS)
The IP Multimedia Service Switching function (IM-SSF)
The Media Resource Function (MRF),
The IP Short Message Gateway (IP-SM-GW)*

Működési modell összevetése a tesztelési fázisokkal



Követelmények implementálása, MT és FT tesztelése entitás szintjén, hibák riportolása

Végső hálózati verifikáció

Szolgáltató specifikus tesztelés

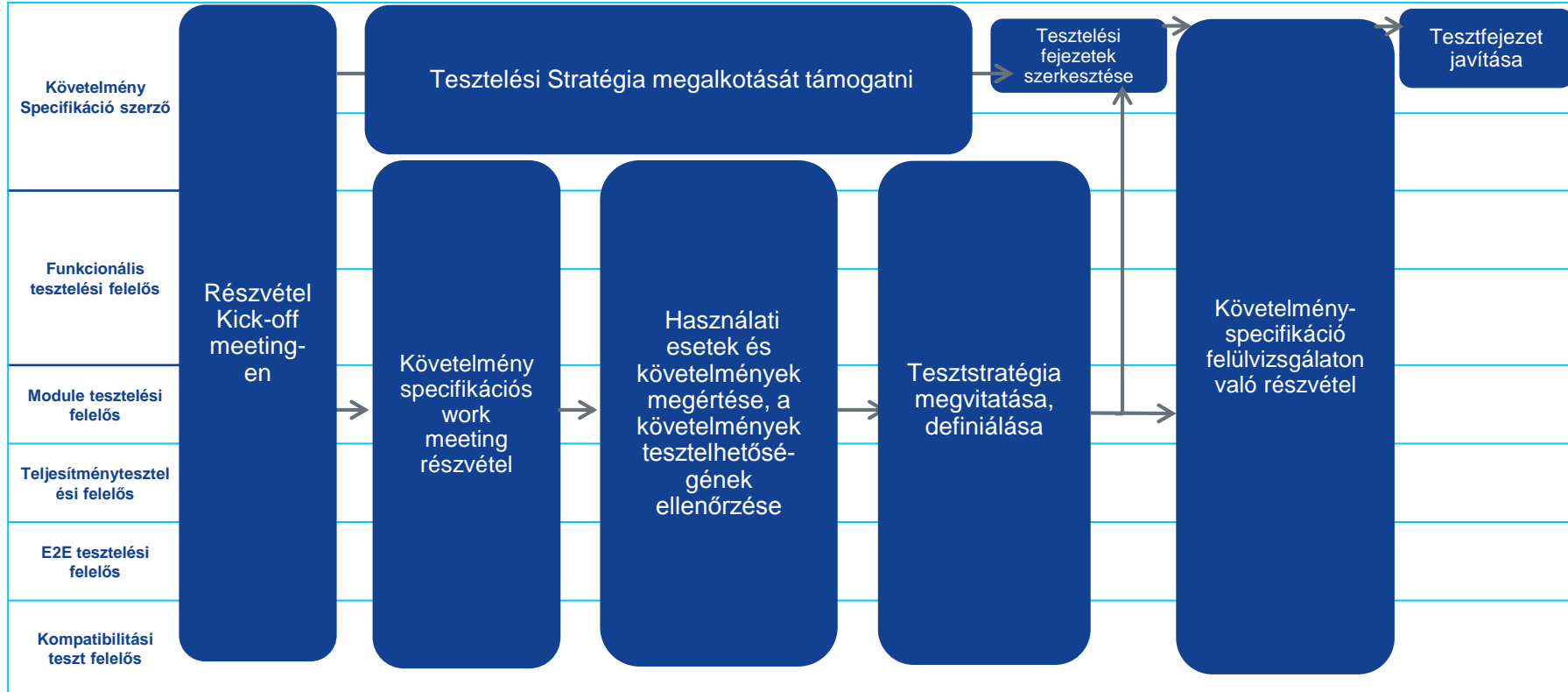
„Commercial Use”

t=0

Folyamatos hálózati verifikáció (E2E funkcionális és non-funkcionális szempontból)

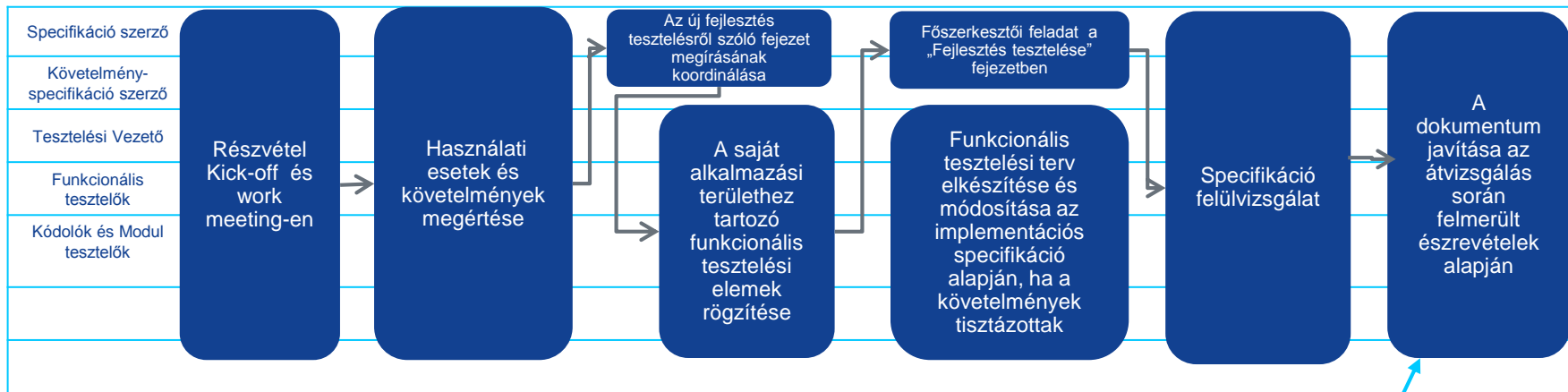
Követelmény Specifikációs fázis

Leggyakoribb akciók



Implementáció specifikáció fázis

Leggyakoribb akciók



Kapcsolódó metrika: Specifikáció meeting részvételi ráta

Modul tesztelés

- Unit teszt ritka, ezért gyakorlatilag ez tekinthető az első tesztelési fázisnak
- MT terv bevezetve, sima text fájl, struktúrája a következő.
 - REFERENCE: melyik folyamatban lévő fejlesztéshez tartozik, vagy hibajegyhez
 - KEYWORDS: legfontosabb releváns kulcsszavak
 - PURPOSE OF THE TESTCASE: mit tesztelünk
 - EXECUTION OF THE TESTCASE: releváns szekvenciaelemek
előzetes konfiguráció
 - EXPECTED RESULTS: mit várunk el
 - AUTHOR: Testcase készítője
 - BRANCH: Release info
- Saját gépen új tesztek , MT regressziós teszt futtatása automatizáltan

Adott fejlesztési szelet(ek) modul tesztelése kész → kód átvizsgálása

Metrikák:

- Kód felülvizsgálatának fedettsége
- Modul teszt fedettség a meglévő kódra
- Modul teszt fedettség az új implementációra

Funkcionális tesztelés

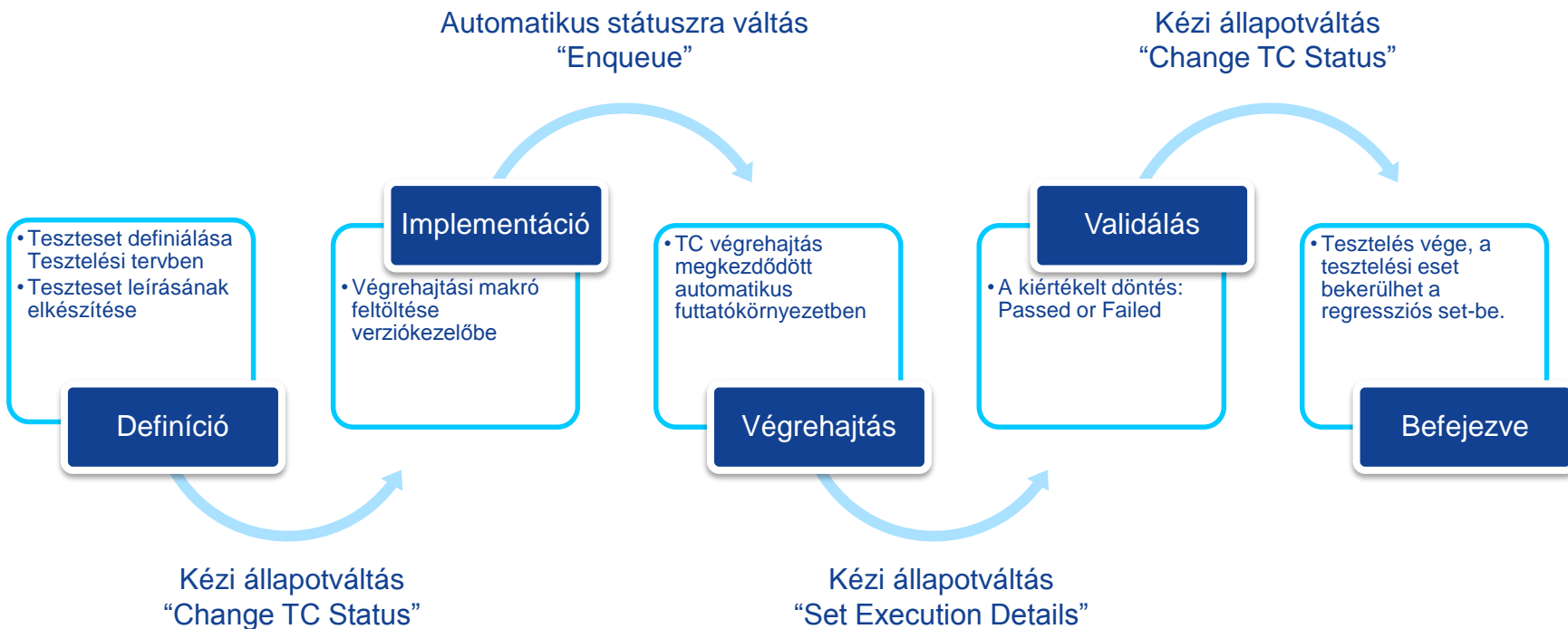
- Az életciklus több szakaszában, többféleképpen
 - Új fejlesztések
 - Regressziós
 - Karbantartási

Funkcionális tesztelés új fejlesztéseknél

- Tesztelési Terv XHTML formátumban tárolt
- **Definition of Done tartalmazza pl.:**
 - ✓ Összes tesztet felülvizsgálattal lezárva
 - ✓ Tesztelési tapasztalatok leírása
 - ✓ Átadás egy adott szakterület funkcionális tesztelőinek
 - ✓ Dokumentálni: hogyan kell az új fejlesztés funkcióit aktiválni, használni



Funkcionális teszteset életciklusa



Validációs döntések a Funkcionális Tesztelés szintjén



Passed

- A tesztet végrehajtási eredménye megfelel a leírásban lefektetett elvárt eredménynek



Failed

- A tesztet végrehajtási eredménye eltér az elvárt eredménytől, hibát találtunk



Blocked

- Hardverhiba, vagy hibás SW konfiguráció blokkolja a tesztet végrehajtását



Postponed

- Egy folyamatban lévő új fejlesztés vagy hibakorrekció miatt a végrehajtás elhalasztásának lehetősége

Regressziós tesztelés (FT)

Tesztelési módok és azok szkópja:

1. **Időkorlátozott (Time-boxed)**: Előre definiált időablakban, limitált erőforrásokkal végrehajtva, a lényeg: ,durva', gyors visszajelzést adni a szoftver minőségéről.
 2. **Teljes**: a lehető legmagasabb szoftverminőséget elérni, kiértékelve az összes regressziós tesztet.
- Használati esetek:
 - Szoftverfrissítésnél
 - Új fejlesztésnél
 - Release szintű
 - Karbantartási
 - Ütemezése: Lehet egy adott sprintre, de hosszabb távra is

Regresszió futtatás mérőszámai (példák)

- Completeness kiértékelt tesztvégrehajtások % -os aránya
- Maturity sikeresen lefutott tesztvégrehajtások % -os aránya
- Automatic Execution Error Ratio (AEER) azon sikertelenül lefutott tesztvégrehajtások száma, ami a végrehajtó környezet hibájából származik / összes végrehajtott regressziós teszteset száma
- EDD(Escaped Defect Density)
$$\frac{\text{új fejlesztési ágból származó hibák száma}}{\text{specifikált, implementált, letesztelt tesztelési kondíciók}}$$
- RPE (Requirement Production Efficiency)
$$\frac{\text{új fejlesztéssel töltött mérnökórák száma}}{\text{specifikált, implementált, lefedett tesztelési kondíciók}}$$

Rendszer Verifikációs fázis

- Az alábbi tesztelési fázisokat különböztetjük meg
 - **Platform rendszertesztelés, PfST**
 - **Végponttól-Végponttig terjedő funkcionális verifikáció, e2eFV**
 - **Üzemeltetési és karbantartási Tesztelés (O&M)**
 - **Teljesítmény verifikáció, PeV**
 - **Kompatibilitás tesztelés, CT**
 - **Release Upgrade Tesztelés**
 - **Biztonsági Tesztelés**

- **Módszer: Folyamatos integráció (Continuous Integration, CI)**

Folyamatos integráció, CI

- A folyamatos integráció minden tesztelési fázis előkövetelménye
- A cél, hogy minél gyorsabb visszajelzést adjunk a build-ekben történő fejlesztésekről, javításokról (Agile -> gyakori kód „merge” (GIT))
 - Hetente visszajelzés, a kódnak mindig a vevőhöz szállításra kész állapotban kell lennie
- MT szint: Fontos megbizonyosodni arról is, mikor készült el az összes tervezett funkció, ami által funkcionális tesztelésre küldhető
- Az Applikációs build-eket különböző platformokon is ellenőrizni kell
 - PfST CI az entitások szintjén végrehajtva: a rendszer leginkább hibaérzékenyebb területeire próbál fókuszálni, minden build-ben egy kis sanity set ennek szól. Minőség fenntartása miatt figyelembe vesszük az előző release-ben tapasztaltakat
 - PeV CI az entitások szintjén:
 - Konfigurációs és Forgalmi Profilok létrehozása, ügyféligényekhez mérten, a kapacitást is figyelembe véve, egy specifikus release-re
 - E2E CI hálózati szinten végrehajtandó
- CI Tesztesetek:
 - rendszer újraindítások,
 - CPU Switchover
 - A Unit állapotának megváltoztatása
 - Diagnosztikai futtatások
 - Forgalmi mérés
 - Stabilitási mérés

Network Bring-Up

- Cél:
 - a rendszertesztelésben résztvevő összes hálózati elem/generátor integrálása, és konfigurálása
→ az új funkciók verifikálása megkezdődhet
 - Bring-up szükséges a teszt tool-okra, terminálokra, kliensekre
 - E2E és PeV esetén is

Platform Rendszertesztelés, PfST

- Alkalmazásréteg: a konkrét termékre jellemző üzleti logika (TAS-ban pl. hívásirányítás, hangtovábbítás)

<-> Platformréteg: a hozzá kapcsolódó, alkalmazástól független támogató szolgáltatások, melyek közösek a termékekre nézve : pl. adatbázis-kezelés, helyreállítás, licenzkezelés, üzenetkezelés

- Minden, az adott release-ben megjelenő új platformfejlesztést tesztel
- Középpontban a provokatív tranzakciók és a nem-funkcionális területei a rendszernek
- Kifejezetten a rendszerszintű üzenetekre fókuszál a teszteredmények vizsgálata során
- Kísérleti/tapasztalat alapú tesztelést alkalmaz

E2E Funkcionális Verifikáció, e2eFV

Cél:

- a legújabb rendszerszintű funkciók, új programkomponensek végponttól végpontig történő tesztelése, valós környezetben, valós eszközökkel
 - A követelmények az FRS-ben és a Rendszerszintű Architectúra Specifikációban rögzítettek

Kliens teszteszközök

- Nokia Communication Suite: Skype-szerű
- Többféle VoLTE-képes terminállal való tesztelés:
 - Qualcomm,
 - Samsung S4,S5
 - iPhone (Bundle)

Verifikációs és validációs eszközök, példák vizsgálatokra

- Wireshark: SIP-szekvencia
- Statisztikai report: elindultak-e a szolgáltatások
- Codec-ek egyeztetése sikeres volt-e (WB-AMR)
- Connection Pool: összerendel két telefont, és megvizsgálja, hogy kiépült a beszédút



Üzemeltetési és Karbantartási Teszt

- Alapvető vizsgált funkciók:
 - Szoftverkonfiguráció adminisztrálása
 - NetAct hálózati menedzsment szoftver segítségével szolgáltatók ellenőrizhetik, irányíthatják és optimalizálhatják hálózatukat.
 - Adatkapcsolati Hálózati Funkciók (TCP/IP and OSI) ellenőrzése
 - Biztonsághoz kapcsolódó műveletek

Teljesítmény Verifikálás, PeV

- Megnövelt forgalomra és teljesítményre koncentrálni, illetve a tesztelt hálózati elemet a normál kondíciókat meghaladva is vizsgáljuk (túlterhelés, hibatűrés)
 - Megnézi az egyes hálózati elemek közötti együttműködéseket is (MGW, HLR/HSS)
 - Kulcsfontosságú tényezők: időtartam, terhelés, forgalmi tényezők (hívások, üzenetek, handoverek, LocUp-ok száma)
 - Előre konfigurált hálózati elemeken
 - **SUT valós, a többi szimulált**, de azokat is előre kell konfigurálni
 - Előre generált forgalmi profilokkal (amik az ügyfelek forgalmi profiljaira alapulnak)
- PI. Regisztrációk tesztelése: Többféle azonosítása lehet a felhasználóknak, és ezeknek működni kell (TEL-URI, SIP TEL-URI,stb.)

•Alfínusok•

Összehasonlító teszt	SW / HW összehasonlítás
Stabilitás teszt	A rendszer legalább akkor forgalmi intenzitású terhelés alatt áll, mint amekkorára tervezve lett.
Túlterheléses teszt	Megvizsgálni, hogy bírja a hálózati elem a túlterhelést, hogy tudja átvinni a forgalmat, hogyan hat rá.
Provokatív teszt	Hogyan tud megbirkózni a hálózati elem a különböző rendhagyó/véletlen szituációkkal, karbantartási műveletekkel, load alatt.
Kapacitás teszt	Különböző részletes mérések (pl. különböző hívások, Location Update-ek)
Komponens tesztelés	Olyan újabb fejlesztések vizsgálata, amiket a Stabilitás teszten kívül is jó lenne külön tesztelni

Stabilitás tesztelés

- A tesztelt hálózati elemen akkora intenzitású forgalmat hajtunk végre, amit el kell bírnia.
 - Sokrétű, megnövelt forgalom (e.g., többfajta hívás) teljesen felkonfigurált hálózati elemen.
 - Egy adott release legfontosabb funkcióinak forgalmi alapú tesztelése
 - Mérések sokasága lefut
 - Számlázáshoz kapcsolódó adatok átvitele
 - Tesztelés időtartama: 12-24 óra: Több millió hívás egyszerre.
 - Követelmény: ne bukjon hívás, miközben a felek beszélgetnek egymással
- Célok:
 - Hibákat / nem elfogadható hibariasztásokat észlelni
 - A SW-package stabilitását ellenőrizni(Nincs unit újraindulás, Nincs switchover)
 - Ellenőrizni kell, hogy a kereskedelmi használatra belőtt kapacitástervezés szerint, a CPU terheltség nem haladja meg a 70%-ot

Túlterheléses tesztelés

- Megvizsgálni, hogy a hálózati elemek hogyan viselkednek a túlterhelés hatására, hogyan engedik át a forgalmat
- Minden Computer Unit akkora forgalmat kap, hogy minimum 150%-os terhelés legyen.
- Cél:
 - Ellenőrizni, hogy a hálózati elemek stabilok és képesek kezelni a forgalmat túlterhelt körülmények között is.
 - Pl. Memóriahasználat: 5%-os memóriaelterésnél át kell nézni a teljes rendszert, ki „eszi meg” a memóriát.
 - A túlterhelés ellenőrző mechanizmus csökkenti a forgalmat

Provokatív tesztelés

- Megvizsgálni, hogy a tesztelt hálózati elem hogyan küzd meg különböző abnormális helyzetekkel, és hogyan orvosolja őket, bizonyos futtatási körülmények között
- Ilyenek pl. a unitok újraindítása, switchover, az egész hálózati elem újraindítása
- Max. 70%-os CPU terheltség mellett
- Maximális forgalom mellett történik a provokáció
- A cél:
 - Verifikálni, hogy minden hálózati elemben egy normál karbantartói mechanizmus működik
 - Verifikálni, hogy a hálózati elemek tudnak abnormális helyzeteket kezelni
 - Verifikálni, hogy a unit redundancia segít a switchover kezelésében
 - Verifikálni, hogy a rendszer feláll jósolható időn belül.
- Vizsgálati példa / hibaeset
 - Pl. Log-olásra használjuk a disk-et, majd egyszer csak elfogy a hely → Programblokk/modul újraindulhat

Kapacitás tesztelés

Példák jellemzőkre:

	Darabszám (nagyságrend szerint)
VLR-ben lévő előfizetők száma	10^6
Mobil BHCA forgalom	10^6
Kimenő SMS-ek (óránként)	10^5
Bejövő SMS-ek (óránként)	10^6
Egyidejűleg fennálló hívások maximális száma (óránként)	10^5

(BHCA = **B**usy **H**our **C**all **A**ttempts): Processzorok forgalomkezelő képessége, egy forgalmas óra során kezelhető hívások maximális számában megadott jellemző

Kompatibilitás tesztelés

- A cégszintű kompatibilitási szabályzat alapján
- Hálózatba helyezett új elem kompatibilitása a többi, régebbi elemmel
- Három legfőbb cél
 - Azokat a legfőbb hibákat megtalálni, amiket a hálózati elemek közötti interface implementáció rejt:
 - Szabványosítási szempontból
 - Specifikációtól való eltérés szempontjából
 - A megfelelő funkció ellenőrzése végfelhasználói szempontból, valódi hálózati elemekkel és terminálokkal
- Többféle hívás eset: 2G - 3G - PSTN - ISDN , többféle szolgáltatást kipróbálva
- Ügyfelek Release Kompatibilitási Riportot megkapják, ha befejeződött

LC/E2EVe – Szoftver Release Frissítési Teszt

- Minimális (kb. 10%) load szint mellett
- Ügyfelek számára biztosítani korábbi release-ről átállni, zökkenőmentesen, az újabb szoftverszintre való váltás
- Különböző, ügyfelek számára támogatott upgrade-
"útvonalak" az egyes termékekre

Tipikus mérések PeV-en

- Forgalmi mérés
 - Virtuális áramkörök csoportjaira
 - Forgalmi kategóriák
 - Cella
 - SMS
 - UPD
 - IP ATM hálózati forgalom
- Jelzésátvitel mérése
 - MAP interfész
 - CAP interfész
 - MEGACO interfész
 - SIP
- Mobilitás mérése
 - VLR terheltség
 - Handover
- Terheltség megfigyelése
 - Virtuális áramkörökre
 - Computer Unitokra
 - Ethernet Message Bus-ra
 - Sikertelen hívások megfigyelése
 - Teljes leterheltség mérése
- Egyéb riportok
 - Elérhetőség
 - Forgalom-teljesítmény
 - Hibaüzenet kódok (Jelzésátvitel specifikusan)
 - Szolgáltatások mérése
- Stb.....

PeV tesztelési eszközök

- Ahhoz, hogy különböző load-ot generáljunk a tesztelendő hálózati elemekre, **forralomgenerátorok** alkalmazunk

A hiányzó hálózati elemekre szimulátorokat alkalmazunk

Időzítések alkalmazás arra, mintha telefonról indulnának a hívások.

Pl.

- MAP-generator (MAP-protocol)
- LTE-generator (Diameter-protocol)
- SMS-generator (TCP/IP)
- GU-generator (BSSAP, BSSAP' and RANAP-protocols),
- BICC-generator (BICC-protocol)
- Megaco-generator (H.248-protocol)
- SIP-generator (SIP-T, SIP-I, SIP, SIP-UA)
- SCP-simulator (INAP+CAP+SINAP protocols)

→ IPSL scriptnyelvre átültetve

- 3rd party eszközök: Pl. Polystar Solver (IU-CS over IP/ATM, SIP-UA, AoIP))

- NSN termékek és a statisztikák monitorozására

•Egyéb eszközök

Billing center szimulátor

WireShark (EtheReal)

Biztonsági tesztelés

Cél: Alapvető biztonsági problémák detektálása

- Portszkenelés
 - Nyitott portok keresése, melyik szolgáltatások futtatják őket
- Robusztusság tesztelés
 - Nagymennyiségű, automatikusan generált, szándékosan rosszul formázott protokollüzenet küldése a cél felé
- Sebezhetőségi tesztelés
 - Specifikus minták kimutatása, amik a sebezhetőség meglétét kimutatják
- Denial of Service (DOS) Tesztelés
 - Az erőforrások elérhetetlenné tételé DOS támadással
- Lehet teljes rendszerhez és külön termékhez kapcsolódó

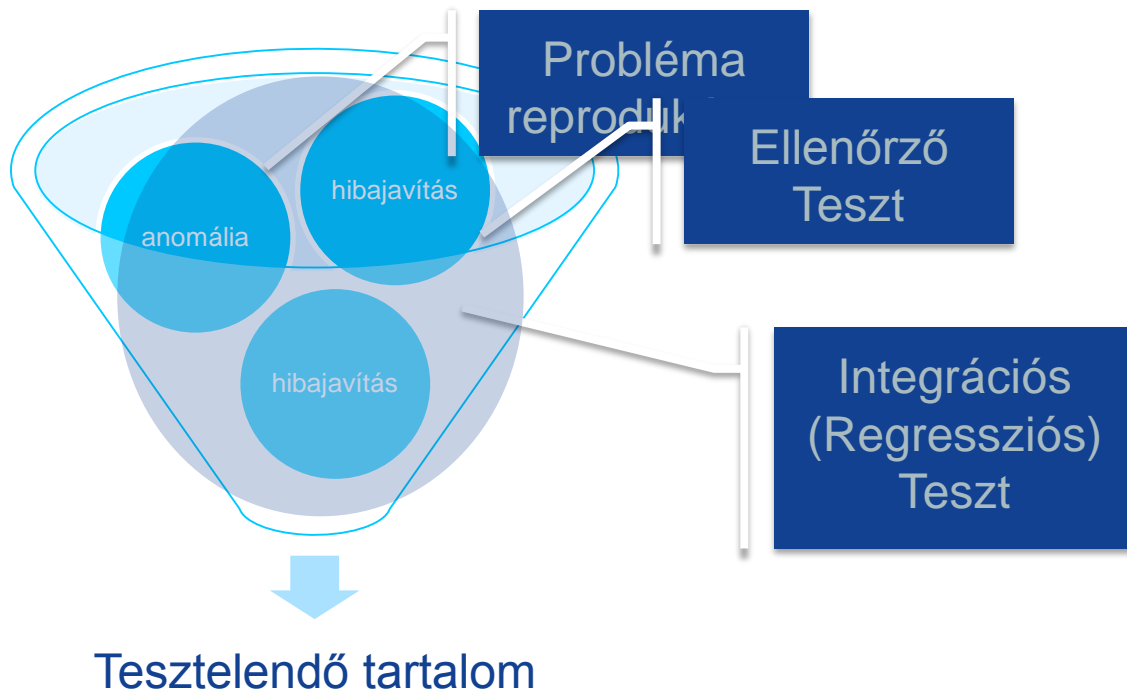
Karbantartási tesztelési elemek

Metrikák:

- Defect riportok
 - Belső: szervezeten belüli
 - Külső: ügyfeleknél
- Egy hibajegyet 24 órán belül konkrét személyhez kell rendelni

Eszközök:

- RCA/EDA



?