

Alerant alkalmazásfejlesztői keretrendszer

Az Alerant Zrt. nagyvállalati informatikai rendszerek fejlesztésekor szerzett tapasztalataira építve dolgozta ki alkalmazásfejlesztői keretrendszerét. Az AFF a fejlesztés teljes életciklusában segítséget nyújt valamennyi szereplő számára, ehhez egy javasolt alkalmazáscsomagot, fejlesztési módszertant és számos saját kiterjesztést tartalmaz. A keretrendszer használata felgyorsítja a fejlesztés folyamatát azáltal, hogy átláthatóvá és menedzselhetővé teszi azt, valamint egységesíti a fejlesztési tevékenységet nagyméretű, heterogén, elosztott környezetben is.

Szerver oldali alkalmazásfejlesztői keretrendszer



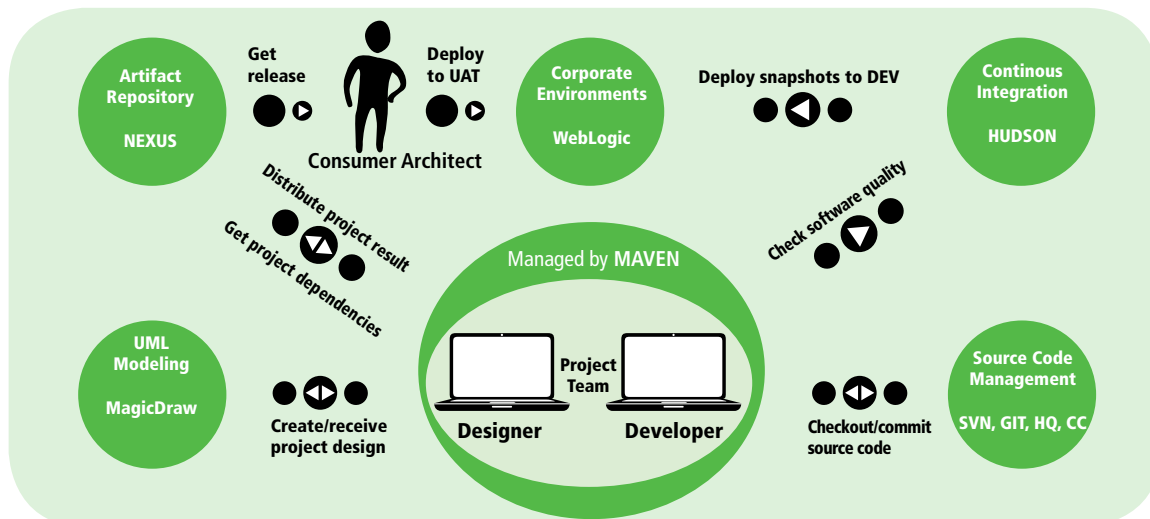
A keretrendszer átfogó megoldást nyújt a fejlesztés teljes életciklusában. Három fő elemből épül fel:

- fejlesztési folyamatok
- támogató folyamatok
- infrastruktúra

Alkalmazáscsomagunkat a tapasztalataink szerint legmegbízhatóbban és legjobb hatékonysággal működő komponensekből állítottuk össze, a kapcsolódó módszertan pedig a tervezéstől a kiadásig nyújt gyakorlati támogatást az alkalmazások optimális felhasználásához.

A kialakított referencia-architektúra a komponensekhez javasolt implementációt adja, míg a saját kiegészítések lehetővé teszik e komponensek még hatékonyabb illesztését a kidolgozott fejlesztési módszertanhoz. A módszertan rögzíti a fejlesztésben résztvevők lehetséges szerepköreit, felelősségi körüket, valamint elvárt tudásuk szintjét. Javaslatot tesz a kiosztott szerepkörök alapján a projektagok munkamódszerére, illetve a teljes fejlesztési folyamatra is.

A megoldás referenciaarchitektúráját az alábbi ábra szemlélteti:



Az így létrejött fejlesztési keret az alábbi előnyöket biztosítja:

- **A tervezés és a fejlesztés szoros kapcsolata**

A tapasztalatok azt mutatják, hogy jelentős problémát okoz az alkalmazás tervezésének és a megoldás implementációjának elcsúszása. A fejlesztési fázisban a készülő alkalmazás olyan technológiákat/megoldásokat tartalmazhat, amelyek nem megbízhatóak és nélkülözik a megismert legjobb gyakorlatokat. A probléma feloldására egy MDA-alapú (modell-vezérelt architektúra, model-driven architecture) megközelítést alkalmazunk, mely nem csak a legjobb gyakorlatok meglétét garantálja, hanem technikai keretet is nyújt a készítendő alkalmazások számára. Mindezt a tervezés során kialakított modellből állítja elő, ami garantálja az alapvető strukturális, felépítésbeli igényességet.

Az MDA további előnye, hogy lehetővé teszi a rendszer különböző absztrakciós modelljeinek egységes kezelését, valamint segítségével az absztrakciós szintek könnyen átjárhatóak lesznek. Az MDA lehetővé teszi a feladat hatékony dekompozícióját, vagyis több részre bontását és elemzését is. Az UML-modellből egy androMDA elnevezésű MDA-implementációval generáljuk a modellezett elemek vázait vagy bizonyos esetekben annak egészét. Saját kiegészítésekkel generálhatunk JPA-entitásokat, EJB 3-as elemeket, Web Service-eket de akár POJO-kat vagy standard Exception-öket. Nem csak atomi elemeket hozhatunk létre, léteznek template-ek különböző tervezési mintákra, legjobb gyakorlatokra is.

- **Kódminőség garanciák**

A megoldásban rögzített módszertan teljes kiépítésével kódminőségi elvárások is érvényre juttathatók. A projektagok tudás-szintjéből adódó eltérések kimutathatók és kiegyenlíthetők, így a módszer használatával garantálható, hogy az elkészített kódok mind formailag, mind tartalmilag megfelelnek a cég által kialakított elvárásoknak. Az elkészített kódok vizsgálatát formailag a Checkstyle, tartalmilag pedig PMD-eszközökkel vizsgáljuk, míg az egész rendszer felépítésének mutatóit a JDepend eszközzel tesszük láthatóvá. Mindez a Continuous Integration szerver összevont dashboardján lesz elérhető. A folyamatos integrációval, a változásfigyeléssel a problémák időben előtérbe kerülnek.

- **Validálható használati eszközkészlet**

Fontos szempont a fejlesztések során, hogy ne kerüljön olyan külső függőség a készülő alkalmazásba, amely később akár minőségi, akár jogi akadályok miatt a szállítandó alkalmazást tönkreteszi. Igény tehát, hogy az elkészítendő alkalmazásainkba csak olyan validált, minőségileg elfogadott eszköz kerüljön, amit a cég jóváhagy. Az architektúrában szereplő Artifact Repository, illetve a módszertanban meghatározott használat a garancia az elvárás teljesülésére. Az Artifact Repository ezen túlmenően biztosítja,

hogy egyazon problémára a teljes cég ugyanazt a komponenset használja, így biztosítja az elemek újrafelhasználását, valamint az egységes környezet kialakítását. A fordítás során tehát a fordítóeszköz (Maven) csak olyan függőségeket képes feloldani, amelyek megtalálhatók a cég repository-jában.

- **Hitelesíthető átadási folyamat**

Az átadási folyamat rögzítése, megismételhetősége, a verziók kezelése alapvető elvárás az alkalmazásfejlesztés során. Ennek kialakítása minden projektben extra ráfordítást igényel. Általánosításával időt és pénzt takaríthatunk meg, valamint ha a generális folyamat egy hitelesített üzleti folyamatba is ágyazódik, akkor további minőségi garanciákat kaphatunk. A keretrendszer komponenseket tartalmaz a fenti problémák orvoslására. A javasolt Artifact Repository rendelkezik azon képességgel, hogy a telepített artifactokat először egy staging repository-ba tegye, továbbá megadhatunk workflow-kat arra vonatkozóan, hogy a staging repository-ból hogyan kerülhet egy artifact az élő repository alá. A workflow-ban különböző szerepkörű felhasználók képesek az artifact állapotokat megváltoztatni.

- **Könnyebb fejlesztői újrahasonosítás**

Mivel a módszertanban leírt szerepkörök jól beilleszthetők minden projektbe, illetve a használt eszközkészlet állandó, ezért nem igényel folyamatos technológia-betanutást minden projekt kezdetekor, így a fejlesztők újrahasonosítása, projektek közti mozgatása is egyszerűbbé, hatékonyabbá válik. Lehetővé teszi továbbá, hogy a fejlesztők az üzleti problémákra fókuszáljanak a technológiai nehézségek helyett.

- **Gyors fejlesztés**

Az MDA-alapú megközelítés segíti, hogy az elkészített rendszertervek alapján megszülessen a projektstruktúra, létrejöjjenek a komponensvázak és a technológiai konfigurációk. A tervezésre fordított energia a fejlesztési időben így még jobban megtérül. A fejlesztők a technológiai nehézségek megoldása, az új technológiák illesztése helyett az üzleti logikára tudnak koncentrálni, mert a többit a keretrendszer szolgáltatja. A keretrendszer frissítéseivel automatikusan életbe lépnek az újabb technológiák és a meglévők hibajavításai is.

A keretrendszer tehát számos alkalmazásfejlesztés során felmerülő problémára nyújt megoldást, így téve még hatékonyabbá a cég produktivitását, emellett biztosítja a minőségi elvárások betartását is.

Kliens oldali alkalmazásfejlesztői keretrendszer – WEB GUI

Az Alerant web GUI fejlesztői keretrendszerének technológiai alapját a Sun által szabványosított és a J EE részét képező JSF-keretrendszer adja. Ennek segítségével korszerű, komponens alapú webes felületek fejlesztése valósítható meg.

Mivel az alap JSF-komponensek gyakran nem elégítik ki a megrendelők egyre kifinomultabb igényeit, ezért szükség volt egy Ajax alapú, JSF-en nyugvó keretrendszer kiválasztására. Az ICEFaces-ben szabványos JSF-komponensek találhatók az Ajax keretrendszerbe implementálva, ami a mai kor webes alkalmazásai előtt álló kihívásokra teljeskörű választ ad. Segítségével olyan interaktív felhasználói felületek készíthetők, amelyek már-már a vastag kliens technológiával készített alkalmazások tudásával vetekszenek.

A szofisztikált pageflow logikát igénylő alkalmazások esetén Spring Web Flow-t alkalmazunk. A JSF-keretrendszerbe ágyazott Spring Web Flow technológia révén a fejlesztők munkája hatékonyabbá válik, mivel egyrészt lehetővé teszi a pageflow-k újrahasonosítását sub flow-k definiálásával, másrészt a képernyők mögött használt adatok hatókörének megválasztásához szélesebb körű lehetőséget biztosít, amelyekkel az összetett, wizard jellegű képernyők fejlesztése rövidebb idő alatt, és nem utolsósorban kevesebb egyedi megoldás alkalmazása nélkül valósítható meg.

A felhasználók által bevitt adatok integritásának ellenőrzésére saját validációs keretrendszert használunk, aminek alapját az Apache Commons Validation csomagja adja. Ezzel a keretrendszerrel deklaratíván tesszük az ellenőrzési szabályok kezelését, amivel a szabályok újrafelhasználhatóvá válnak az alkalmazás architektúrájának különböző rétegeiben. Ezzel a megoldással felgyorsul a fejlesztés és a validációs szabályok kezelése.

A portlet alapú felhasználói felületek készítésekor az inter-portlet kommunikáció megvalósítása igényli az egyik legnagyobb odafigyelést és ráfordítást. Ehhez a feladathoz az ICEfaces Ajax Push technológiáját használjuk, valamint az Oracle Coherence Grid megoldásával biztosítjuk a kommunikáció során kezelt adatok tárolását és továbbítását az – akár elosztott környezetben futó – portletalkalmazások között.

Fejlesztői eszközkészlet

Fejlesztéseink során szigorú szabályokat állítunk fel a szoftver projektstruktúrára. A projektek build életciklusát támogatja az Apache Maven eszköz. Az Apache Maven által nyújtott lehetőségek kihasználásával előre meghatározott szabványos könyvtárszerkezettel dolgozunk, így tesszük felépítésüket tekintve egységessé projektjeinket. Az egységesség biztosítására maven archetype-okat készítettünk, amelyekkel a projektek elkezdéséhez szükséges konfigurációs feladatok a minimumra csökkennek. A fejlesztők munkáját eljárásgyűjtemények, valamint saját fejlesztésű UI-komponensek teszik hatékonyá a keretrendszerben. Facelet alapú kompozit komponenseink elsősorban a web űrlapok készítését teszik egyszerűbbé. Ezekben a feladatokban rendszerint először egy kérdést teszünk fel a felhasználónak, aki válaszol arra. Ha szükséges, a kérdés értelmezését tooltip segítségével tesszük érthetőbbé, majd a választ ellenőrizzük. Amennyiben az ellenőrzés sikertelen, hibaüzenetet írunk ki a felhasználói felületre. Normál esetben ennek a feladatnak a megvalósítása több elemi komponens használatát igényli, amiket minden egyes kérdés esetén a fejlesztőnek újra és újra be kell gépelnie. Kompozit komponenseink ezt a struktúrát fogják össze egyetlen komponensben, újrafelhasználható módon.

A pénzügyintézetek és telekommunikációs cégek számára különböző, szektorspecifikus, kompozit komponenscsomagokat készítettünk. Banki projektjeinkben ennek köszönhetően – például egy bankszámlaszám bekérése esetén – a fejlesztőnek nem kell a beviteli mező maszkolásának és ellenőrzésének megvalósításával foglalkoznia, mert azt az erre készített komponens már tudni fogja. A kompozit komponensek ezen felül az egységes felhasználói felület megvalósításában is hathatós segítséget nyújtanak. A fejlesztők számára az Ajax alapú webes felhasználói felületek tesztelésére is biztosítunk eszközkészletet. A funkcionális tesztekhez a Selenium eszközt használjuk, aminél a teszt scriptek készítését fejlesztői dokumentációval és mintascriptekkel segítjük. A teljesítményteszt készítésére komplex tesztsomagot állítottunk össze. Ennek alapját az Apache JMeter eszközkészlete és saját fejlesztési keretrendszer biztosítja. Ezzel a komplex Ajax Push technológiát használó felhasználói felületek tesztelése válik egyszerűbbé és gyorsabbá.

A fejlesztések minőségének ellenőrzésére szabványos eszközöket használunk. A programozói hibák kiszűrésére a FindBug-ot és PMD-t, melyekkel rendszeresen ellenőrizzük a kód minőségét. Az egységes stílus és kódformázás támogatását pedig a CheckStyle eszközzel biztosítjuk. Ez utóbbival az új fejlesztők betanulási idejét tudjuk lecsökkenteni, az átlátható, egységes stílus alapján formázott forráskódok révén.

Az Alerant kapcsolódó szolgáltatásai

Az Alerant átfogó támogatást nyújt a megoldás bevezetésében, a fejlesztési igények és folyamatok felmérésétől, a referencia-architektúra telepítésén és a fejlesztési módszertan átadásán keresztül a bevezetés-támogatásig. Igény szerint, a bevezetés után további folyamatos támogatást biztosít.



Alerant Informatikai Zrt.
Infopark sétány 1.
1117 Budapest
Telefon: +36 (1) 205-0055
Fax: +36 (1) 205-0056
E-mail: info@alerant.hu
<http://www.alerant.hu>

Az Alerantról

Az Alerant Zrt. a hazai nagyvállalati informatikai piac meghatározó tagja. Ügyfelei számára teljes üzleti megoldásokat nyújt. Tevékenységi körébe J EE-alapú alkalmazások fejlesztése, integrációja, bevezetése, valamint támogatása tartozik. Ügyfelei nagyvállalatok, ezen belül elsősorban távközlési cégek, pénzügyintézetek, iparvállalatok, valamint államigazgatási intézmények, így referenciái között szerepel az Erste Bank, a Magyar Telekom, valamint az AstraZeneca, a MOL és az Állami Foglalkoztatási Szolgálat. Az Alerant a Deloitte közép-európai piacot vizsgáló Fast 50 felmérésében 2008-ban negyedik, 2006-ban a „Rising Star” kategóriában első helyezést ért el. 2008-ban a Deloitte EMEA-térséget vizsgáló Fast 500 felmérésben 27. lett.