SZAKDOLGOZAT

Jagusztin László

2014
IT projektek szakmai termékeinek hasznosulása

a projekt zárása után

Készítette: Jagusztin László
Informatika menedzsment szak
2014

Szakszeminárium-vezető: Dr. Fehér Péter
Témavezetői nyilatkozat

Alulírott, ................................................................. konzulens kijelentem,

hogy a fent megjelölt hallgató fentiek szerinti szakdolgozata (egyetemi/ mesterképzésben
diplomamunkája)  **benyújtásra alkalmas és védésre ajánlom.**

Budapest, . .................................

.................................................

(a konzulens aláírása)
II. számú melléklet

NYILATKOZAT

A SZAKDOLGOZAT NYILVÁNOSSÁGÁRÓL

Név (nyomtatott betűvel): Jagusztin László

Szakirányú továbbképzés, képzés neve: Informatika menedzsment szak

Egyéb képzés neve:

Dolgozatom elektronikus változatának (pdf dokumentum, a megtekintés, a mentés és a nyomtatás engedélyezett, szerkesztés nem) nyilvánosságáról az alábbi lehetőségek közül kiválasztott hozzáférési szabályzat szerint rendelkezem.

TELJES NYILVÁNOSSÁGGAL

A könyvtári honlapon keresztül elérhető a Szakdolgozatok/TDK adatbázisban (http://szd.lib.uni-corvinus.hu/), a világháló bármely pontjáról hozzáférhető, fentebb jellemzett pdf dokumentum formájában.

KORLÁTOZOTT NYILVÁNOSSÁGGAL

A könyvtári honlapon keresztül elérhető a Szakdolgozatok/TDK adatbázisban (http://szd.lib.uni-corvinus.hu/), a kizárólag a Budapesti Corvinus Egyetem területéről hozzáférhető, fentebb jellemzett pdf dokumentum formájában.

NEM NYILVÁNOS

A dolgozat a BCE Központi Könyvtárának nyilvántartásában semmilyen formában (bibliográfiai leírás vagy teljes szöveges változat) nem szerepel.

Budapest, 2014 Május 1.

..................................................

a szerző aláírása
Tartalom

Tartalom........................................................................................................................................6
Bevezetés .........................................................................................................................................8
  A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?..........................10
  Mely projekt szereplők feladata ezeket elkészíteni? ...............................................................11
  A szakmai termékeknek a fejlesztési folyamat mely fázisában kell elkészülniük? .............13
  Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében? ..........14
Projekt definíció..........................................................................................................................15
  „A projekt fogalmai: ................................................................................................................15
  „A projekt tulajdonságai: .........................................................................................................15
  Projektirányítás definíciója ......................................................................................................17
  Projektvezetési módszertan ....................................................................................................18
Projekt indítás..................................................................................................................................20
  Kulcstermék a PAD („projektalapító dokumentum”): .........................................................20
  Projekt indító dokumentumok: ...............................................................................................20
  Projekt irányítás termékei: .....................................................................................................20
Befektetés indokolás .....................................................................................................................21
Szerkezet .......................................................................................................................................24
  Szállító kiválasztás, szerződés ..............................................................................................24
  Licensch feltételek biztosítása .................................................................................................25
  Forráskód, tulajdonjog tisztázása..........................................................................................26
  Támogatási szerződések megkötése ....................................................................................26
  Szavatosság, garancia ..............................................................................................................27
  Karbantartás, támogatás .........................................................................................................27
  Üzemeltetés ................................................................................................................................28
  PRINCE2 projekt szervezet ....................................................................................................30
Tervezés .........................................................................................................................................33
Projekt szakaszok........................................................................................................................36
  Követelmény kezelés szakasz ...............................................................................................38
  Rendszertervezés szakasz.......................................................................................................42
  Implementációs szakasz, szoftver fejlesztés ..............................................................................45
Implementációs szakasz, rendszer építés .................................................................50
Tesztelés szakasz ........................................................................................................52
Oktatás szakasz ..........................................................................................................63
Projekt zárás ...............................................................................................................66
Összefoglalás ...............................................................................................................70
Ábrajegyzék ................................................................................................................72
Szakirodalom lista .......................................................................................................73
Internetes források jegyzéke ......................................................................................74
Bevezetés

Arra keresem a választ, egy IT projektnek milyen szakmai termékeket kell előállítania, hogy a leszállított IT rendszer hosszútávon fenntartható módon, gazdaságosan üzemeltethető legyen. Minden IT projekt információt állít elő: adatokat (pl. migráció), program kódotok, dokumentációkat. Ezen információk előállításának a megtérülése lényegesen függ attól, hogy előállításuk mennyire az egyedi projekt tevékenységéhez kapcsolódik, vagy szerencsés esetben újra felhasználásra tudnak kerülni a termék teljes életciklusa alatt.

Az elterjedt projekt menedzsment módszertanok, mint például a Prince2 vagy a PMBOOK részletes segítséget adnak a projektvezetők számára, hogy a projektek:

- Milyen modell szerint épüljenek fel
- A projekt irányításához szükséges szervezet milyen legyen
- Hogyan határozzuk meg a projektvezetési feladatokat
- Milyen dokumentumokat, projekt termékeket állítsunk elő a projektben

Az elterjedt projekt menedzsment módszertanok, mint például a Prince2 vagy a PMBOOK részletes segítséget adnak a projektvezetők számára, hogy a projektek:

- Milyen modell szerint épüljenek fel
- A projekt irányításához szükséges szervezet milyen legyen
- Hogyan határozzuk meg a projektvezetési feladatokat
- Milyen dokumentumokat, projekt termékeket állítsunk elő a projektben

A projektek jelentős mennyiségű erőforrást használnak fel a működésükhez szükséges és a projekt céljául szolgáló végtermék előállításához. IT projektek esetén a projekt fázisok során:

- Követelmény specifikáció
- Tervezés
- Fejlesztés
- Tesztelés
- Oktatás
- Migráció
- Üzemeltetésre való átvétel

nagy mennyiségű dokumentáció készül, ami szükséges a projekt fázisok dokumentálásához és a következő fázisok számára bemeneti információkat szolgáltatnak.

Dolgozatomban azt kívánom vizsgálni, hogy ezen dokumentumok életútja hogyan változik a projekt bevezetése után: egyszeri erőfeszítés-e a létrehozásuk, ami az IT
termék előállítását szolgálja csak, vagy van arra lehetőség hogy hasznosuljanak a termék életciklusának további szakaszaiban is az üzemeltetés és továbbfejlesztés során. Ha ezzel a feltételezéssel élünk, akkor pedig hogyan lehet olyan módon elkészíteni ezeket a projekt termékeket, hogy ne csak a projekt rövid távú céljait, hanem a termék életciklus hosszú távú céljainak is megfeleljenek.

A dolgozatomban szándékosan nem vizsgálom a projekt saját irányítási és minőség biztosítási termékeit, mert ezek a projekt lezárásával nem részei az elkészült terméknek, ezért a termék további életciklusában nem vesznek részt.
A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?

Az IT projektek az általuk előállított dokumentációkat, kódokat, leszállítandókat a projekt fázisai alapján mindig a következő fázis igényeinek megfelelően állítják elő. Jellemző dokumentációk a

- Követelmény specifikáció
- Megoldás terv
- Funkcionális specifikáció
- Technikai specifikációk
- Logikai Architektúra terv
- Fizikai Architektúra terv
- Rendszerterv
- Teszt terv
- Telepítési leírás
- Felhasználói dokumentáció

A fenti dokumentációk mind a projekt végtermékének az előállítását és az üzemeltetésre való átvételét segítik elő, a projektben meghatározott szerepkörök számára történnek leszállításra.

Dolgozatomban azt kívánom elemezni, hogy egy általános vállalati szervezetben kiknek kell szólnia ezen dokumentációknak és a projekt lezárása után milyen felhasználási területe lehet ezen projekt termékeknek. A felhasználási területek ismerete lényegesen segíti, hogy a megfelelő minőségben és részletezettséggel készüljenek el a projekt szakmai termékei. Természetesen a projekt korlátai: idő, források, cél, rendelkezésre álló szakértelmi és erőforrások korlátozásával és tökéletes minőség soha nem tud készülni, azonban ha meg tudjuk mutatni, hogy a projekt szakmai termékei hol és hogyan térülnek meg, lehetőséget teremtünk arra, hogy a projekt céljait kiszélesítsük vagy a megtérülését jobban tudjuk bizonyítani, így a rendelkezésre álló források mennyisége növelhető.
Mely projekt szereplők feladata ezeket elkészíteni?

Az elterjedt projekt menedzsment módszertanok, mint például a Prince2 vagy a PMBOOK részletes segítséget adnak a projektvezetők számára, hogy a projektek szervezete hogyan épüljön fel és a projekt szervezeten belül mely szerepköröknek milyen dokumentációkat kell elkészítenie.

A projekten belül típusok szerint készülhetnek:

Irányítási termékek, amelyek konkrétan a projekt irányítás célját szolgálják:
- Szervezeti felépítés
- Működési szabályzat
- Előrehaladási jelentések
- időtervek
- Erőforrástervek
- Emlékeztetők

Minőségirányítási termékek:
- Minőségellenőrzési jelentések
- Quality Gate-ek
- Audit jelentések

Szakmai termékek, IT projektek esetén például:
- Követelmény specifikáció
- Megoldás terv
- Funkcionális specifikáció
- Technikai specifikációk
- Logikai Architektúra terv
- Fizikai Architektúra terv
- Rendszerterv
- Teszt terv
- Telepítési leírás
- Felhasználói dokumentáció
Dolgozatomban azt kívánom elemezni, hogy ezen projekt termékek előállítása egy IT projekt esetén mely projekt szervezeti egység feladata és mely szervezeti egységek bevonásával kell elkészülnie egy általános vállalat esetén.
A szakmai termékeknek a fejlesztési folyamat mely fázisában kell elkészülniük?

Az IT projektek számos terméket állítanak elő, (kód, dokumentáció) amelyek a projekt folyamata során egymásra épülnek, az egyes szakaszok szerint. A projektvezetőnek azonban jelentős szabadsága van abban, hogy a projekt termékek előállítása során mikor és milyen módon vonja be a projekt szervezet szereplőit, illetve a megrendelő szervezetek képviselőit.

Ennek jelentős hatása van a minőségre, illetve az egyes követelmények hangsúlyára az elkészült végtermék tekintetében. Jellemző módon a követelmények nagy része nem áll olyan pontosságban, minőségben és mennyiségben a projekt indításakor rendelkezésre, hogy egy soros folyamaton keresztül megvalósítható legyen a termékfejlesztés, mindenképpen valamilyen iteráción keresztül finomodnak a gyakorlatban.

Dolgozatomban ahol lehet elemezni fogom, hogy egyes projekt fejlesztési módszertanok szerint a projekt dokumentációik/termékeknek a projektek mely szakaszában kell elkészülniük, és milyen projekt fázisokban milyen iterációkon keresztül pontosodnak.
Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?

Ha nem a projekt lokális optimumot vesszük figyelembe, hanem a vállalati folyamatok integráns részének tekintjük ezeket a dokumentumokat, akkor az IT projektek számára a vállalat hosszú távú céljainak is megfelelő termékek tudnak készülni elkerülve azt, hogy egy az adott projektre egyedi dokumentáció halmaz készüljön, ami nem szolgálja sem a projekt végtermékének az életciklusát sem a vállalati folyamatok támogatását. Azonban jól megfogalmazva a vállalat hosszú távú érdekeit elősegítő követelmény rendszert, az IT projekteket a megfelelő irányba lehet befolyásolni ezen cél elérésé érdekében.

A dolgozatom további részében egyenként elemzem egy IT projekt termékeit, a fenti 4 szempontrendszer alapján. A Prince2 projekt módszertant használatom sorvezetőnek, de az projekt által előállított IT termékek nem kapcsolódnak egyetlen módszertanhoz sem szorosan, ezért nem függenek tőle, önmagukban is érdemes vizsgálni őket. Ez megadja a szabadságot ahhoz, hogy bármely módszertan szerint működő projekt fel tudja használni ezt az elemzést. Ahol releváns, meg fogom említeni az egyes módszertanok közötti különbséget.
Projekt definíció

A dolgozatom témája a projekt, ezért röviden definiálnám:

„A projekt egy időben behatárolt erőfeszítés egy egyedi termék, szolgáltatás vagy eredmény létrehozása céljából. „
(Projekt Menedzsment Útmutató PMBOK® Guide, 2006)

„Projektnek tekintünk minden olyan egyszeri és egyedi komplex feladatot, amely egy előre meghatározott eredmény (cél) elérésére irányul meghatározott idő- és költségkereten belül. „
(Görög Mihály, 2008 - Projektvezetés)

„A projekt fogalmai:
- Idő
- Költség
- Tevékenységek
- CÉL
- Erőforrás
- Minőség „
(Langer Tamás, 2007 – Projektmenedzsment a szoftverfejlesztésben)

„A projekt tulajdonságai:
- Véges és határozott időtartamú
- Előre megfogalmazott és ellenőrizhető végtermékek előállítása a cél
- A termékek előállításához szükséges összefüggő tevékenységek hálója
- Előre rögzített erőforrások
  Szervezeti struktúra, pontosan megfogalmazott feladat és hatáskörökkel, amely a projekt irányítását lehetővé teszi
- A projekt, mint feladat egyedi
- Átfogó, komplex
Az IT projektek természetesen ennél szűkebben is definiálhatóak, mert az egyedi termék, szolgáltatás vagy eredmény egy adott szerverezet IT céljait valósítja meg. Dolgozatomban ennél is szűkebben vizsgálok meg a témát, a szolgáltatást és eredményt nem vizsgálok, csak ha a termék egy információrendszer.

(Molnár Bálint, 1997 – Projektirányítás módszertana (PRINCE))
Projektirányítás definíciója

PRINCE 2:
A projektmenedzsment a projekt valamennyi szempont szerinti megtervezése, követése és ellenőrzése, és mindazok motiválása, akik részt vesznek a projekt céljainak az előre meghatározott időben, költségkereteken belül és megfelelő minőségben történő megvalósításában.

PMBOK®:
A projektmenedzsment a projekt tevékenységeinek végrehajtása során a tudás, a képességek, az eszközök és a technikák alkalmazása a projekt követelményeinek teljesítése céljából.

Dolgozatomban szándékosan nem vizsgálom a projekt irányítási termékeit. Az oka az, hogy témám a projekt szakmai termékeinek a hasznosulásának a vizsgálata a projekt befejezése után, a projekt irányítási termékei viszont nem részei a szakmai termékeknek amelyek részt vesznek az IT termék életciklusában:

![Diagram](image)

1. ábra: IT termék életciklus (Ficsor Lajos, Dr. Kovács László, Krizsán Zoltán, Dr. Kusper Gábor – 2009, Szoftvertesztelés)
Projektvezetési módszertan

Dolgozatom témája a projekt szakmai termékeinek a hasznosulása a projekt lezárása után. Ezért alapvetően nem kötődik egyetlen projektvezetési módszertanhoz sem, bármelyik esetén alkalmazható az elemzésem. Azonban egyet kiválasztottam, hogy sorvezetőnek használjam a projekt termékek elemzéséhez. Ez a Prince2 módszertan volt, mert ez állt legközelebb a témához:

A PRINCE 2 által használt projekttervezés termék alapú, azaz a tervezés az átadandó termékekhez összpontosít, nemcsak egyszerűen a különböző tevékenységek időzítését adja meg.

(Langer Tamás, 2007 – Projektmenedzsment a szoftverfejlesztésben)

A Prince 2 összetevői a következők:

![Diagram showing Prince2's components](image)

A fenti struktúra alapján a következő fejezetben csoportosítom a projekt termékeit és a szakmai termékeket a bevezetésben meghatározott 4 dimenzió alapján fogom elemezni:
• A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
• Mely projekt szereplők feladata ezeket elkészíteni?
• A szakmai termékeknek a fejlesztési folyamat mely fázisában kell elkészülniük?
• Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?
Projekt indítás

Kulcstermék a PAD („projektałapító dokumentum“):
- Célok
- A projekt határai
- Feltételek
- Felhatalmazások
- Erőforrások, ütemezés

Projekt indító dokumentumok:
- A minőségi naplót
- Az esemény-naplót
- A projekt tapasztalatait, tanulságait rögzítő naplót
- A következő szakasz tervét

Projekt irányítás termékei:
- Kezdeti és záró jelentés
- Adott szakasz aktualizált terve (terv/tény összehasonlítás)
- A következő szakasz terve
- Felülvizsgált projekt-terv
- Javított kockázati napló (risk log)
- Tanulságok listája
- Minden egyéb változás dokumentálása a személyzetben vagy a projekt vezetésében

A fenti projekt indító és projekt irányítás elősegítő termékek nem szakmai termékek, a projekt zárása után nem szoktuk elővenni. A célok meghatározása alapja lehet egy későbbi visszamérésnek, hogy mennyire sikerült ezeket elérni, azonban ezt a PRINCE2 szerint a projekt lezárásakor meg kell tenni, ezért ezt sem elemzem, mert a szakmai termék életciklusának nem része.
Befektetés indokolás

„The Business case presents the optimum mix of information used to judge whether the project is (and remains) desirable, viable and achievable, and therefore worthwhile investing in.” (Managing Successful Projects with PRINCE2, 5th ed)

Tartalma:

- Vezetői összefoglaló
- Projekt indítás okai
- Üzleti alternatívák
- Várható haszon
- Várható hátrányok
- Időterv
- Költségek
- Befektetési igény
- Kockázatok

Magyarosan csak BC-nek hívott befektetés indokolási terv szorosan nem szakmai termék, nem része az elkészült IT rendszernek. Azonban alapja a termék megtérülési számításának és a termék életciklusát is végigköveti (jó esetben), ezért részletesebben elemezem.

- A projekt folyamat mely fázisában kell elkészülnie?
  A BC készítés közvetlenül a projekt indítás után felfadat. A BC ahogy a PRINCE2 módszertanban is látható az ábrán folyamatos frissítés alatt van a projekt szakaszok alatt és a projekt zárás fázisban pedig kiértékelésre kerül.
3. ábra, Business Case (OGC, Managing Successful Projects with PRINCE2, 5th ed)

- Mely projekt szereplők feladata ezeket elkészíteni?
  Mivel projekt szervezet még nem létezik, egy erre dedikált csapat feladata ez, akiket a projektvezető irányít. Rendkívül fontos, hogy a BC elkészítésében minden olyan vállalati szervezet részt vegyen, amelyek számára az adott projekt költséget generál. Tipikus probléma, hogy a BC költség oldala sokszor nem teljes, mert kimaradnak szervezetek / költség elemek belőle, vagy nem terjed ki a termék teljes életciklusára (pl. 3 év támogatással a beszerzési árban vásárolunk eszközöket, a 4. évtől jelentkező támogatási költséget pedig már nem számoljuk) így a megtérülés i számítás nem a valóságot tükrözi.
  Ezt csak szabályozással lehet megelőzni, ahol a BC készítést vállalati szinten szabályozzuk, felsorolva a kötelező elemeit.

- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
  A BC mindenképpen a felső vezetés vagy projekt szponzor számára készül.

- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?
  A BC készítés legnagyobb problémája, hogy a projekt létezésén lényegesen túlnyúhat a projekt megtérülése. A projekt zárása után viszont megszűnik az a szervezet amelyik létrehozta és frissítette. A PRINCE2 javaslat szerint egy „Benefit review plan”-t érdemes létrehozni, ami a projekt zárás után is tartalmazza a teendőket és ezt egy erre szakosodott vállalati szervezett
számára át kell adni. Így a BC készítésbe fektetett erőforrások a projekt zárása után is megtérülhetnek.

Érdekes kérdés az új agilis projekt vezetési módszertanok hogyan közelítik meg ezt a kérdést. Szerencsére amit a PRINCE2 még csak ajánlatként fogalmaz meg, azt az agilis projekt menedzsment már kötelezővé teszi, a Sprintek szerves része a BC folyamatos frissítése és visszacsatolás a vezetés felé, ezzel lehet kezelni hogy az agilis projektekben jellemzően nagy bizonytalansági faktort a lehető legkisebb pénzügyi kockázattal kezeljük:

![4. ábra, Agile Investment (The Business Case for Agile, 2011 - SolutionsIQ.)](image-url)
Szállító kiválasztás, szerződés

Az első dolog ami feltűnhet az ábrán az a szállító. Az üzlet és a felhasználó természetszerűleg létezik a projekt indításakor, de a szállító nem biztos, őt ki kell választani és szerződést kell vele kötni. A következő szerződés típusok és kiválasztási eljárások létezhetnek:

<table>
<thead>
<tr>
<th>Tradicionális szerződés</th>
<th>Nyilt versenyeztetés</th>
<th>Szelektív versenyeztetés</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kulcsrakész szerződés</td>
<td>Szelektív versenyeztetés</td>
<td>Kétlépcsős versenyeztetés</td>
</tr>
<tr>
<td>Menedzsment-szerződés</td>
<td>Kétlépcsős versenyeztetés</td>
<td>Meghívásos versenyeztetés</td>
</tr>
</tbody>
</table>
A szerződés kötés része a scope és pénzügyi elszámolási kérdések tisztázása mellett a következő IT szakmai termékek létrehozása:

**Licensz feltételek biztosítása**

A legtöbb informatikai projekt nem nulláról fejleszti ki az általa létrehozni kívánt információrendszert, hanem számos meglevő komponenstre építkezik (operációs rendszerek, adatbázis kezelők, alkalmazás szerverek, fejlesztési keretrendszerek) amelyek gyártói licenszbeli rendelkezhetnek, illetve maga a felhasználó által használt alkalmazás is lehet licenszbeli rendelkező termék amit a projekt testre szab.

- A projekt folyamat mely fázisában kell elkészülnie?

Ezeket a licensz szerződéseket érdemes a projekt megvalósítási szerződésével egy időben tisztázni és a pontos felhasználási feltételeket és kötelezettségeket rögzíteni. A projekt megvalósítási szakasza alatt a licenszbeli konstrukciók közvetlenül befolyásolhatják az architektúrális és funkcionális terveket, mert jelentős költségvonzatuk lehet. Ezen kívül a teljes termékciklust végig kísérik ezért jelentős hatásuk van a megtérülésre. A licensz költségek tisztázása és a költségviselő jó megválasztása a projektnek ebben a szakaszában sokat segíthet az optimális tervezési döntések meghozatalában.

- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?
- Mely projekt szereplők feladata ezeket elkészíteni?

Érdemes a licensz konstrukcióhoz igazítani azt a költségviselőt, akinek közvetlen hatása van rá (pl. felhasználó alapú licenszek: Üzlet, technológiai licenszek: IT) aki így megfelelő irányba tudja befolyásolni a tervezést.
Forráskód, tulajdonjog tisztázása

A projekt megvalósítási szerződés része a forráskód és tulajdonjog tisztázása is. Szállító által végzett külső fejlesztés esetén a szerzői jogok kérdését tisztázní kell, mert a szellemi alkotások szerzői jogával a szerzők élhetnek. Az ide vonatkozó jogi szabályozást az 1999. évi 76. (a szerzői jogról szóló) törvény, 58-63. § adj meg. Ez kimondja, hogy a szoftvert már megcsületése pillanatától a szerzői jogi védelem megilleti. A szoftverre vonatkozó vagyoni jogok átruházhatóak és a felhasználónak joga van a szoftverről egy biztonsági másolatot készíteni.

A szoftverek forráskódja és a dokumentációk a fejlesztők önálló szellemi alkotása. Amennyiben belső fejlesztésről van szó, akkor a legtöbb esetben ha egy céghez felvesznek alkalmazottként egy programozót, akkor a szerződésben külön szerepel hogy mindennemű szellemi termék a cég tulajdona, így ez nem probléma.

Ha nem belső fejlesztésről van szó, akkor egy szoftverhez úgy juthatunk hozzá ha megvesszük a licenszét: felhasználási jogot vásárolunk. Fontos, hogy ez nem jelent tulajdonjogot.

A licensz konstrukciók nagyon sokfélek lehetnek: Perpetual, Shareware, Freeware, Open Source, GNU, GPL.. a dolgozatomnak nem ez a fő témája ezért részletes ismertetést ezekről nem írok.

A projekt zárása után kiemelt jelentősége van annak, hogy milyen lehetőségekkel rendelkezünk a továbbfejlesztésre, javításra: hozzá vagyunk kötve egy szállítóhoz vagy magunk dönthetjük el hogyan folytattuk a projektet. Amennyiben rendelkezünk a forráskóddal, lehetőségünk van a rugalmas szállító váltásra. A hosszú távú karbantartási és továbbfejlesztési költségek alacsony tartásának a kulcsa az, hogy a kezünkben legyen az IT termék forráskódja és jogunk legyen változtatni azon szabadon, így megnyilik a lehetőség a versenyeztetésre az egyes projekt zárás utáni feladatok esetén.

Támogatási szerződések megkötése

- A projekt folyamat mely fázisában kell elkészülnie?

A támogatási szerződés első gondolatra a projekt zárási tevékenységgel függ össze, ebben a szakaszban kellene végezni. Azonban az, hogy milyen típusú támogatási
szerződést kötünk, jelentősen meghatározhatja a szállító érdekelt ségét azzal kapcsolatban, hogy az elkészült termék mennyire lesz dokumentált, üzemeltethető, a fenntartásának a költség szintje mennyire lesz magas. Ezért érdemes a szállító kiválasztással egy időben a támogatási kérdésekkel kapcsolatos alapvetéseket tisztázni és rögzíteni.

- Mely projekt szereplők feladata ezeket elkészíteni?
- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?

Mint mindig itt is érdemes azon projekt szereplőket bevonni ebbe a tevékenységbe ahol a kockázat, felelősség vagy költség felmerül ezzel kapcsolatban. Az IT Üzemeltetés képviselője vagy a Szolgáltatás menedzser az, aki ezt a leg hatékonyabban tudja képviselni.

- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?

Szavatosság, garancia

- Hibajavítás az átadás után
- Jogszavatosság

A szavatosság, garancia kérdése alapvető a megtérülés szempontjából. Ha tűl hosszú periódust kötünk ki vagy tűl kemény feltételeket, a szállító ezt be fogja árazni és olyanért fogunk fizetni, ami nincs arányban a valós kockázattal. Ha nem kérünk ilyet, akkor viszont a szállító nem fogja olyan minőségben szállítani a terméket ami elvárható. Érdemes olyan konstrukcióit választani, ami a valós kockázatokat tükrözi és a projekt indításakor rögzítésre kerül, mert ez biztosítja hogy a projekt összes szakmai terméke az ennek megfelelő minőségben fog a szállító részéről elkészülni: érdekében áll majd neki is a minőség kontroll.

Karbantartás, támogatás

- Megbízási konstrukció
- Szolgáltatási szintek
- Kötbérek
• Teljesítés ellenőrzésének módja

A karbantartási, támogatási szerződés a garanciális időszak után lép életbe. Az alapvető kérdéseit: konstrukció, szolgáltatási szint, kötbér lehetőség, riportok, auditok lehetősége érdemes ugyancsak a szállító kiválasztással egy időben megtenni. Az a szállító aki nem gondolkodik hosszú távon, korlátos méretű a helyi támogatási szervezete, nincs megfelelő a megrendelő anyanyelvén beszélő szakembergárdája, stb. nagyban növelheti a termék teljes életciklusának a költségeit (TCO).

Üzemeltetés

A szállító sok esetben üzemeltetési szolgáltatást is vállal a projekt befejezése után.

Első gondolatra ez nagyon optimális megoldásnak tűnik, mert a termék biztosan az elvárt szolgáltatási szintnek megfelelően fog elkészülni (amellett hogy divatos és modern dolog volt az elmúlt években az outsourcing), de számos problémát vet fel:

- A megfelelő szolgáltatási szint szerződés (SLA) megkötése komoly megrendelő oldali jogi, gazdasági és IT ismereteket feltételez, ami sok esetben nem áll rendelkezésre

- Ha az üzemeltető outsourcing partner tevékenységét részletesen akarjuk szabályozni, akkor részletes szolgáltatás meghatározási szerződést (OLA) is kötnünk kell, ami még komolyabb szakmai ismereteket feltételez a megrendelő részéről és nagyon erőforrás igényes lehet az elszámoltatása

- Folyamatos outsourcing menedzsment tevékenységet kell végeznünk, aminek a költsége hozzáadódik a szolgáltatás költségéhez

- Az elkészült IT termék nem a vállalat igényeinek, hanem az SLA/OLA szerződésnek megfelelően fog működni. Szerencsés esetben ez persze egybeesik, de a rugalmassága mindenképpen korlátozott

- Az elkészült IT termék nem a vállalat hanem az outsourcing partner szempontjainak megfelelően készül el. Szerencsés esetben ez is egybeesik, de a sok szereplő megnehezíti az érdekek érvényesítését.
Összefoglalva a támogatási szerződések jelentős mértékben befolyásolják az IT termék megtérülését a teljes életciklusa alatt (TCO), ezért ezt a szempontot a megfelelő szakmai szervezeteknek képviselnie kell a szállító kiválasztás fázisában.
PRINCE2 projekt szervezet

A szállító kiválasztás lépéseiivel a PRINCE2 módszertan nem foglalkozik részletesen, alapvetésnek veszi a szállító létezését. Véleményem szerint a projekt szervezetet mindenképpen két lépcsőben kell kialakítani: először a szállító nélkül, amíg a kiválasztás és pályáztatás szakasza zajlik, majd a klasszikus módszertan alapján teljessé tenni.

A projekt szervezetben az üzleti megrendelő és a szállítói érdekeknek mindenképpen meg kell jelenniük. Érdekes, hogy egyetlen projekt vezetési módszertan sem tartalmazza a vállalat hosszú távú szakmai érdekeinek a képviseletét (architektúra, nem funkcionális követelmények, szabványok..) amelyek ugyancsak nagyon fontos szerepet kapnak a projekt szakmai termékek hosszú távú megtérülésében.

![Diagram](image)

6. ábra, Projekt szervezeti struktúra, (OGC, Managing Successful Projects with PRINCE2, 5th ed)

Még a fenti részletesebb ábrán sem szerepel a vállalati IT szakmai érdekek képviselete. A minőség menedzsment stratégiában persze megjelennek majd ezek a szempontok később, de általában ez már késő: a pályáztatás és szállító kiválasztás folyamatába
sokszor csak az üzleti funkcionalitás kerül bele és gyakran még a követelmény felmérés fázisba sincs bevonva az IT, mert a nem funkcionális követelmények felmérése a gyakorlatban sokszor elmarad.

A másik kérdés a szervezettel kapcsolatban, hogy az adott rendszert ki veszi át és ki fogja működtetni. A felhasználók alatt könnyen csak a valódi végfelhasználókat értjük, pedig egy alapvetően üzleti rendszer esetén is maga az IT szervezet is felhasználó, hiszen az ITIL szerinti folyamatait működtetnie kell:

Amennyiben nagyvállalati IT szervezetről beszélünk, a siló és kompetencia kérdések is előkerülnek: attól függően melyik IT szervezethez kerül üzemeltetésre a rendszer, úgy változhatnak a nem funkcionális követelmények a vállalati kultúra és rendelkezésre álló kompetencia függvényében.

Összefoglalva a projekt szervezetben a „User” fogalmát mindenkiéppen tágabban kell értelmezni és az üzleti felhasználók mellett az IT fejlesztésnek (szabványok, architektúra) és IT üzemeltetésnek (ITIL folyamatok) is meg kell jelennie már a projekt indításakor. Ez biztosítja, hogy a szállító kiválasztástól egészen a projektzárásig az IT nem funkcionális követelmények is képviselve legyenek az adott IT szervezet
érdekeinek megfelelően, biztosítva a projekt szakmai termékeinek legjobb megítélését. Tapasztalatom alapján a legjobb gyakorlat az, hogy a Projekt Assurance szervezette a Szolgáltatás felelöst (aki az adott IT szolgáltatásért felel a vállalaton belül minden tekintetben) és a Szolgáltatás menedzsért (aki az adott IT szolgáltatás napi működéséért felel majd) is delegáljuk, illetve már a projekt indításakor kijelöljük. Ők mindeneképpen abban lesznek érdekeltek, hogy az IT hosszú távú érdekeit képviseljék a projekt időtartama alatt. Az Ő feladatuk első körben az, hogy a vállalati üzemeltetési, fejlesztési, architektúra, biztonsági.. szabványokat és feltételeket átadják a projektvezetőnek, hogy IT szakmai keretet szabjanak a projekt működésének.

Az agilis módszertanok megjelenésével a projekt szervezet is átalakult:

8. ábra, Projekt szervezeti szerepkörök (Agile Change Management Limited, 2013, Comparing the roles from PRINCE2 with agile project management)

Alapvetően a projekt szereplők hasonlóak, a struktúra és hatáskör az ami változott. A projekt felső vezetése (Board, Manager, Senior User) szinte változatlan formában működik itt is, a projekt alsóbb szakmai szervezeteinek a működésében van a lényeges különbség. Az egyetlen és nagyon pozitív változás a projekt felső vezetésében a Technical coordinator megjelenése, ami reményt ad arra, hogy a nem funkcionális és IT szakmai érdekek képviselte erősödni fog ezekben a projektekben, mert ez a felismerés már a módszertan része.
Tervezés

A tervezés a PRINCE2 módszertan terminológiájában a projekt tervezésére vonatkozik, nem a projekt által elkészíteni kívánt termékre. Ebben a fázisban tervezzük meg a projekt előfeltételeit és erőforrás igényét, fogalmazzuk meg a célokat és elkészíteni kívánt termékeket, illetve elkészítjük az idő és függőségi tervet. Alapja a projekt indításban megfogalmazott célok. A tervezési fázisban definiáljuk a projekt további szakaszait is, ami az alapja lesz a további működésnek.

- A projekt folyamat mely fázisában kell elkészülnie?

A projekt vagy program tervet a PRINCE2 szerint a projekt tervezési fázisában kell elkészíteni.

Tartalma:
- A projekt összefoglalása
- Előfeltételek, függőségek, feltételezések
- Termékek és tevékenységek felsorolása
- Minőségi követelmények
- Kockázatok
- Pénzügyi- és erőforrás tervezés
- Tűréshatár

A projekttervek az egész projektre készülnek el, a jóváhagyott terveket termék-mérőfok kövekkel látják el. Ezután szakasztervek készülnek a projekt minden egyes szakaszára. Legvégül részletes tervek készülnek a szakaszon belüli tevékenységek további bontására. Kivétel kezelési tervek is készülnek, amennyiben szükséges.
PRINCE2 módszertan szerint:

9. ábra, PRINCE2 projekt terv (OGC, Managing Successful Projects with PRINCE2, 5th ed)

A projekt tervek ábrázolására sok esetben Gant chart-ot alkalmaznak, mert ez az egyik leg szemléletesebb ábrázolási módszer. Példa:

![Gant chart példa](image)

10. ábra, Gant chart példa

- Mely projekt szereplők feladata ezeket elkészíteni?

A projekt tervet a projektmenedzser készíti, a projektbiztosító csoport közreműködésével, a projekt kezdetén.

- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?

A projekt tervet nem tekintjük szakmai terméknek hanem projekt irányítási termék, ezért részletesen nem elemzem.
Egyetlen projekt szakaszt azonban kiemelnék és röviden áttekintenék, ez pedig az élesbe állás (GoLive). Az élesbe állás időszaka még a projekt zárása előtt történik jó esetben, azonban ez a szakasz élesen elkülönül a termék fejlesztés és implementációs időszak feladataitól, ezért kiemelt figyelem szükséges a tervezésekor.

A PRINCE2 ebben nem ad segítséget, nincs hozzá template sem, mert a termék központú szemlélete miatt a termék elkészülésekor véget ér a módszertan.

Az élesbe állási terv jellemzően tartalmaz üzleti feladatokat:
- Üzlet folytonossági terv (BCP)
- Üzleti termék specifikus aktivitások (pl. reklám kampány indítása)
- Üzleti kommunikációs terv
- Üzleti paraméterezés
- Eszkalációs terv
- Üzembe állítási teszt terv

Illetve tartalmaz IT specifikus feladatokat:
- IT Üzembe állítási forgatókönyv
- Deployment végrehajtási terv
- Technikai paraméterezés
- Visszaállítási terv
- Technikai üzembe állítási teszt terv

Mindkettő tervezését dedikált szakértői csapatnak kell végeznie a komplexitása és magas kockázata miatt, illetve az Üzleti és IT feladatoknak szinkronban kell lenniük.

- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?
A projekt terv alapvetően irányítási termék, nem része a termék életciklusának, a projekt zárás után nincs szerepe, ezért ezt nem elemzem a megtérülését.
Projekt szakaszok

Dolgozatom további részében projekt szakaszonként elemzem az IT projekt szakmai termékeit a bevezetőben megfogalmazott 4 szempont szerint. A projekt szakaszokban zajlik a szakmai munka, aminek az eredménye a projekt célját megvalósító termékek.

11. ábra, PRINCE2 projekt szakasz kontroll (OGC, Managing Successful Projects with PRINCE2, 5th ed)

A projekt szakmai termékeinek a megtérülése csak a projekt további szakaszaiban kontrollálható, ezért kiemelt fontosságú, hogy ezek a szempontok kommunikálva, mérve és ellenőrizve legyenek a projekt vezetés részéről. Ebben kiemelt szerepének kell lennie a projekt minőség biztosításának, amely a szakaszok elején rögzíti az elvárt minőségi követelményeket, a szakaszok végén a szakmai termékek leszállításakor pedig méri, ellenőrzi ezeket. A gyakorlatban az egyes szakaszok végén a kiválasztott szállító átadja véleményezésre és elfogadásra a szakmai termékeket a megrendelő képviselőinek, akik elfogadják vagy véleményezéssel együtt visszaadják a projekt szakasznak további javításra:
A szakaszokat a hagyományos PRINCE2 módszertanban is használt vízesés modell szerint elemzem a dolgozatom következő részében:

![Diagram](image)


Az egyre gyorsabban változó vállalati környezetben azonban felértékelődik annak a szerepe, hogy a bizonytalanságot a célokban, követelményekben hogyan tudjuk kezelni, ennek megfelelően a változás kezelés költsége egyre nagyobb hangsúlyt kap. Ezért az agilis projekt vezetési módszertan kezd teret hóditani olyan projektekben, amelyek esetén a hagyományos vízesés modell csak magas költséggel és átfutási idővel lehetne alkalmazható. Az agilis módszertan gyors ciklusai és rugalmassága segít ezt a problémát megoldani:
Ezért ahol releváns, meg fogom említeni az agilis módszertan előnyeit és különbségeit a PRINCE2 módszertantól a projekt szakmai termékeinek előállításával és megtérülésükkel kapcsolatban.

**Követelmény kezelés szakasz**

A követelmény kezelés szakasz célja a projekt indítási fázisban megfogalmazott projekt célok nagyobb részletezettségi szinten való megfogalmazása és lebontása olyan egységekre, amelyek alapja lehetnek a további konkrét funkcionális és nem funkcionális tervezésnek.

Csak a teljesség kedvéért említem meg, hogy a PRINCE2 nem ad módszertani segédtelet a követelmény kezeléshez. Ezért a PMBOK módszertanból kölcsönzött ábrával illusztrálnám a fő tevékenységeket a követelmény kezeléssel kapcsolatban:
• Követelmények rögzítése
• Követelmény kezelési terv
• Követelmény összefüggés mátrix, készítése, ami alapján a követelmények közötti függőségek jeleníthetőek meg

A projekt folyamat mely fázisában kell elkészülnie?
A követelmények felmérése, összegyűjtése és dokumentálása a követelmény kezelés projekt szakasz feladata, a szakmai munka kezdete.

Mely projekt szereplők feladata ezeket elkészíteni?
Legtöbbször a szállító végzi, az Üzleti és IT szakmai szervezetek bevonásával. Rendkívül emberi erőforrás és időigényes igényes szakasz: interjúk, workshopok és magas képzettségű üzleti elemzők szükségesek hozzá. Jó strukturálással és párhuzamosítással az átfutási idején lehet optimalizálni. Jellemzően a követelmény felmérési fázis végén sor kerül az informatikai vállalási terv elkészítésére is, ami a projekt IT nézetében tartalmazza az idő és erőforrás tervet.

A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
Az IT szakmai követelményeket két részre bonthatjuk:

• Funkcionális követelmények
  Egy rendszer funkcionális követelményei leírják, hogy a rendszernek milyen funkciókkal kell rendelkezni, hogyan kellene működni (Például aktualizálások, lekérdezések, jelentések, kimenetek, adatok, más rendszerekkel való kapcsolat). Ezek a követelmények a fejlesztett
szoftver típusától, a szoftver leendő felhasználóitól függenek. (Szoftverfejlesztés, Ficsor Lajos,Krizsán Zoltán,Dr. Mileff Péter, 2011)

- Nem funkcionális követelmények
  A nem funkcionális követelmények a funkcionális követelményekkel ellentétben nem közvetlenül a rendszer által biztosított specifikus funkciókkal foglalkoznak, hanem inkább a rendszer egészére vonatkozó eredő rendszertulajdonságokra koncentrálnak. (Szoftverfejlesztés, Ficsor Lajos,Krizsán Zoltán,Dr. Mileff Péter, 2011)

Néhány példa a nem funkcionális követelményekre:

- Rendelkezésre állási elvárások
- Katastrófa tűrési elvárások
- Biztonsági elvárások
- Támogatott platformok
- Rugalmasság
- Könnyen használhatóság
- Robosztusság
- Biztonság

A nem funkcionális követelmények egyik legnagyobb kihívása, hogy sok esetben nem írhatóak le számszerűsíthető módon és nehezen ellenőrizhetőek, kérhetőek számon.

- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?
  o A funkcionális követelmények pontos dokumentálása az egész IT termék életciklusát végigkíséri. A tervezési fázisban mindennek a forrása, a későbbi részletes tervek alapja, a megvalósítási fázisban kétség vagy nem pontos specifikáció esetén is ide lehet visszanyúlni. A teszt esetek alapja is ez az átvételi fázisban. Amennyiben hibát találunk a teszteléskor, az hogy ez „change request – CR” vagy nem, sokszor a követelményből kiindulva lehet csak megmondani. A projekt zárás után is az egyik legfontosabb alapdokumentuma marad a terméknek: segít a
garanciális problémák értelmezésében, illetve ha a rendszert tovább kell fejleszteni vagy frissíteni kell újabb verzióra. Szerepe a termék életcikluson is túlnyúihat, mert a tartalmának nagy része nem termék specifikus, ezért ha új rendszert kell bevezetni a régi helyett, rendkívül sok erőforrást lehet megtakarítani az újra felhasználásával. Előállítása viszont nagyon emberi erőforrás igényes: interjúk, workshopok és magas képzettségű üzleti elemzők szükségesek hozzá. Ezért fontos tudatosítsani, hogy bár a projekt költségvetésének nagy részét jelenítheti a jó minőségű elkészítése, az egyik legjobb megtérülési mutatóval rendelkező projekt termék, ezért érdemes befektetni ebbe.

- A nem funkcionális követelmények már rendszer specifikusabbak mint a funkcionális követelmények. Egy részük általános (pl. biztonsági megfelelősségi követelmények) más részük a konkrét IT rendszerre vonatkozik (pl. támogatott platformok). Ezért az újra felhasználhatósági is alacsonyabb mint a funkcionális követelményeknek. Ami miatt a projekt zárás után is van jelentősége az az, hogy jelentős mértékben meghatározza az adott IT rendszer költség szintjét. A rendelkezésre állás, katasztrófa tűrés, biztonság.. mind olyan követelmények amelyek szinte a végtelenségig fokozhatóak és ezzel együtt a megvalósításukhoz és fenntartásukhoz szükséges költségek is az egekbe szökhetnek. Ezért fontos, hogy ezek a követelmények a projekt tervezési fázisában rögzítve legyenek, az üzleti területekkel validálásra kerüljenek és költségük transzparensen ki legyen mutatva. A projekt zárás után pedig a termék üzemeltetési és fenntartási költségei levezethetőek legyenek mindig a nem funkcionális követelményekből. Így elérhető, hogy amennyiben költségsőkkel engedésre van szükség, azt transzparensen a követelmények változtatásával együtt lehessen megtenni.
Rendszertervezés szakasz

A rendszertervezés szakaszban készülnek el az elkészíteni kívánt szakmai termék szakmai tervezési dokumentumai. Alapja a követelmény kezelési szakaszban előállított funkcionális és nem funkcionális követelmények. Célja, hogy az elkészüljenek azok funkcionális és nem funkcionális tervek amelyek alapjai lesznek az implementációs szakaszban a konkrét IT termék megvalósításának.

A projekt folyamat mely fázisában kell elkészülnie?

A rendszertervek elkészítése a rendszer tervezés szakasz feladata.

- Mely projekt szereplők feladata ezeket elkészíteni?
  Rendszerszervezők, programtervezők, architektek, technológiai szakértők készítik el.

- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
  Az IT tervezés témája nagyon nagyon sokféle tervezési dokumentumot ölel fel, amit az UML Diagram típusokon keresztül szeretnél példát mutatni:

15. ábra, UML diagrammok (Szoftverfejlesztés, Ficsor Lajos, Krizsán Zoltán, Dr. Mileff Péter, 2011)
A tervezési dokumentumokra példák:

- **Funkcionális tervezés:**
  - Üzleti folyamat specifikációk
  - Rendszertervek:
    - Struktúrák: osztály, komponens, objektum diagramok, adatmodellek
    - Viselkedés: Használati esetek, állapotmodellek
    - Interakciók: Szekvencia diagram, adatfolyam diagramok, interfész specifikációk
    - Felhasználói interfész tervek
    - Rendszer biztonsági terv
  - Funkcionális tesztelési tervek

- **Technológiai tervezés:**
  - Deployment diagramok
  - Fizikai szerver, adatbázis, alkalmazás szerver architektúra
  - Hálózati architektúra
  - Logikai és fizikai kapcsolati mátrix
  - Hardver és szoftver méretezés
  - Infrastruktúra rendszer biztonsági dokumentáció
  - Katasztrófá terv (DRP) terv
  - Infrastruktúra mentési terv
  - Infrastruktúra felügyeleti terv
  - Technológiai tesztelési tervek

A funkcionális tervek alapján készül el az implementációs szakaszban a konkrét IT megvalósítása a projekt szakmai terméknek (jellemzően, kód, szoftver) illetve az az IT rendszer, amin fizikailag működni fog a megoldás.
Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?

A funkcionális tervek az egész IT termék életciklusát végigkísérik. A konkrét kódolás, kódgenerálás alapjai a funkcionális tervezési dokumentumok. Amennyiben hibát találunk a tesztek alatt és a követelmény specifikáció helyes, a funkcionális tervek azok amelyekhez igazodnunk kell. Megtérülésük és újra felhasználhatóságuk annál jobb, minél inkább sikerül modelleket egy modellező eszközben ábrázolnunk és ez alapján generálnunk a konkrét kódokat, ezért egy módosítás vagy hibajavítás csak a modellben vagy esetleg generátorban történő javítást jelent. Nagyon fontos szerepe van a modell alapján generált kódoknak a nem funkcionális követelményeknek a minél költséghatékonyabb megvalósításában (példa: egy naplózási követelmény megváltoztatása, ha csak kódgenerátor fejlesztést jelent, nagyságrendekkel kevesebbe kerül ,mint a teljes rendszeren kézzel átvezetni a változtatást). A projekt zárás utáni hibajavítási vagy továbbfejlesztési igények megvalósításának az erőforrás és költség szintjét is jelentősen tudja csökkenteni ha ez a megoldás jól és konzisztensen be lett tartva a projekt tervezési és implementációs fázisában.

A technológiai, fizikai tervezési dokumentumok a konkrét megvalósított fizikai IT rendszerre jellemző dokumentumok. A megvalósítási/rendszerépítési fázisban kiemelt a szerepük mert ez alapján épül meg a fizikai rendszer. Sajnos az újra felhasználásuk nem jellemző, miután már működik egy rendszer a tapasztalatok szerint ezek a projekt termékek a projektek lezárása után nem kerülnek aktualizálásra a változás kezelési folyamatokon keresztül. Megtérülésüket olyan módon lehet javítani, ha tartalmuk betöltése kerül egy ITIL CMDB-t (konfigurációs adatbázis) megvalósító rendszerbe, így az Incidens, Probléma és Változáskezelési
Implementációs szakasz, szoftver fejlesztés

A projekt szoftver implementációs szakasza a rendszer tervezési szakaszban elkészült funkcionális és nem funkcionális tervekre épül. Célja, hogy a tervek alapján az implementációs szakaszban a tervek alapján elkészüljön a konkrét IT termék, ami megvalósítja a projekt indítási fázisban megfogalmazott célokat.

- A projekt folyamat mely fázisában kell elkészülnie?
  A szoftver implementációt a minden IT projektben jelen levő implementációs szakaszban végezzük el. Célja az adott szoftver termék, dokumentációk, kód.. előállítása.

- Mely projekt szereplők feladata ezeket elkészíteni?
  A szoftver terméket fejlesztők, programozók, program tervezők, dokumentáció készítők készítik.

- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
  A szoftver terméket a fejlesztők az üzleti felhasználók és a későbbi üzemeltető szervezetek számára készítik, akik ugyancsak felhasználói lesznek és dolgozni fognak vele (pl. hibakeresés, karbantartási feladatok), csak nem funkcionális szempontból.

  A szoftver terméknek alkalmasnak kell lennie funkcionális működési követelmények kielégítésére és ezen kívül az ITIL szerinti üzemeltetési folyamatokban való részvételre: incidens és probléma kezelés, változás kezelés..

- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében? A leggyakoribb szoftver fejlesztési termékeket részletesenben elemezve:
Kód és hozzá kapcsolódó dokumentáció:
A projekt zárás után a szoftver implementációs szakaszban elkészült termékek: a szoftver és kód dokumentáció minőségétől lényegesen függ a megtérülés, mert a rengeteg rejtett hibát tartalmazó termék javítása már akkor is veszteséget okoz a megrendelőnek, ha a szállító garanciában végzi el: a kiesett idő és esetlegesen elmaradt üzleti haszon mindenképpen veszteség.

![Diagram](image)

16. ábra, Hibajavítás költséggörbe (Molnár Bálint, A projekt irányítás módszertena, 1997)

Ha a hibákat, eltéréseket a szállító olyan megoldásokkal tudja javítani amelyek alacsony átfutási idővel és erőforrással megvalósíthatóak: például modell alapú dokumentáció és kód generátorok, akkor ez a veszteség mérsékelhető valamennyire. Ez igaz az esetlegesen megváltozott projekt célok, követelmények megvalósítására is: a szoftverfejlesztés változáskezelésének költségszintje jelentősen befolyásolja a termék teljes életciklusára vonatkozó költségeket.

Ennek kiemelt jelentősége van az agilis szoftver fejlesztési módszertan esetén is, ahol a módszertan része a folyamatos változtatás és visszamérés. A szoftver fejlesztésben a változáskezelés képessége fejlődött az elmúlt időben az egyik leggyorsabban, a megváltozott vállalati környezeti körülményekhez történő alkalmazkodás miatt.
Telepítési eljárás és technikai konfigurációs dokumentáció
A telepítési eljárás és konfigurációs dokumentáció a szoftver fejlesztéssel egy időben készül el. Segítségével lehet feltelepíteni az adott projekt termékét: a szoftvert az elkészült IT infrastruktúrára. Várhatóan ez nem egyszeri tevékenység, hanem a termék életciklusa alatt sokszor megismétlődik: fejlesztői, teszt és éles környezetek, IT infrastruktúra változások követése, bővítések... ezért a minősége befolyásolja ezen tevékenységek költség szintjét.

Migráció
Kód mellett induló adatokkal is fel kell töltőnünk a rendszert, használatba vétel előtt. Jellemzően ez egyszeri tevékenység, a további adatokat a felhasználói felületeken vagy a többi rendszerrel kapcsolatban álló interfészen keresztül kapja majd. Ezért a projekt zárás után ennek a terméknek a jelentősége kisebb.

Üzleti paraméterezés, termékek, kedvezmények, folyamati beállítások.
Kód és adat mellett üzleti paraméterekkel, termékekkel, folyamati beállításokkal is fel kell tölteni a rendszert, használatba vétel előtt. Ez lehet a migráció része, vagy külön manuális tevékenység, esetleg egyedi tömeges betöltés. Jellemzően ez egyszeri tevékenység, a használatba vétel után további paraméterezést a felhasználói felületeken végezzük. Ezért a projekt zárás után ennek a terméknek a jelentősége kisebb, rendszeresen használt részének a felhasználói leírásba kell kerülnie.

Jogosultság kezelésű leiérás
A rendszer használatba vételéhez felhasználókkal és jogosultságokkal is fel kell tölteni. Ez jellemezően a migráció része vagy egyszeri tömeges betöltésekkel történik. Jellemzően ez egyszeri tevékenység, a használatba vétel után további jogosultság kezelést a felhasználói felületeken végezzük. Ezért a projekt zárás után ennek a terméknek a jelentősége kisebb.
rendszeresen használt részének ugyancsak a felhasználói leírásba kell kerülnie.

- **Felhasználói dokumentáció**
  A felhasználói dokumentáció írja le a szoftver használatát, a funkcionális specifikációknak megfelelően. A felhasználási területe széleskörű: Üzleti felhasználók, alkalmazás támogatás, helpdesk, tesztelők, oktatók használják, mindenki, aki kapcsolatba kerül a szoftverrel. A projekt zárás után termék teljes életciklusa alatt használatban marad, ezért az elkészítésébe fektetett erőforrások megtérülése magas. Jellemző probléma, hogy a változáskezelési folyamatokból kimarad, ezért egy idő után elavul, aktualitását veszti. Ezen úgy lehet javítani, ha a felhasználói leírás nem (csak) külön dokumentum, hanem a szoftver része, aktív hivatkozásokkal a dokumentációból a szoftverrel és fordítva (Help rendszer része), így a megtérülése tovább javítható.

- **Alkalmazás üzemeltetési dokumentáció**
  Az alkalmazás üzemeltetési dokumentáció az alkalmazás és infrastruktúra üzemeltető szervezetek, esetleg a helpesk számára tartalmaz hibakeresési, hibaelhárítási és rendszeres karbantartási feladatokat. A tesztelési szakaszban is már felhasználásra kerül, ugyanezen okok miatt. A projekt zárás után termék teljes életciklusa alatt használatban marad, ezért az elkészítésébe fektetett erőforrások megtérülése magas. Jellemzően a változáskezelését nem a szállító végzi az átadás után, hanem maguk a felhasználók, üzemeltető szervezetek. A szoftverrel kapcsolatos tudásbázis is gyakori hogy épül, ami az üzemeltetési dokumentációban nem szereplő információkat tartalmazza. Ennek oka, hogy a változáskezelése a dokumentációknak lassú és költséges, azonban az incidens és problémamegoldás folyamatok hatékony végrehajtása megköveteli az aktuális hibamegoldási információk rendelkezésre állását.

- **Felügyeletbe vételi dokumentáció, Alkalmazás üzenetek kezelése**
  Az IT projekt szakmai termékeinek része az elkészült megoldás felügyeleti dokumentációja. Ez egyrészt tartalmazza az
automatikus felügyeleti megoldások alkalmazás specifikus paraméterezését (pl. riasztások, határértékek), ez hasonlóan a konfigurációs leírásokhoz egyszeri tevékenység; másrészt az ezzel kapcsolatos manuális tevékenységeket, ami lehet az alkalmazás üzemeltetési dokumentáció része. Megtérülése az említett dokumentációkkal egyenértékű.

- Mentést kérő dokumentáció, visszatöltési terv
  Az IT projekt szakmai termékeinek része az elkészült megoldás mentési dokumentációja. Ez egyrészt tartalmazza az automatikus mentési megoldások alkalmazás specifikus paraméterezését (pl. gyakoriság, mennyiség, tárolási követelmények), ez hasonlóan a konfigurációs leírásokhoz egyszeri tevékenység; másrészt az ezzel kapcsolatos manuális tevékenységeket, ami lehet az alkalmazás üzemeltetési dokumentáció része. Megtérülése az említett dokumentációkkal egyenértékű.
Implementációs szakasz, rendszer építés

A projekt rendszer implementációs szakasa, a rendszer tervezési szakaszban elkészült funkcionális és technológiai tervekre épül. Célja, hogy a tervek alapján az implementációs szakaszban a tervek alapján elkészüljön a konkrét IT infrastruktúra, amin a szoftver implementációs fázisban elkészült termék működni tud majd.

- A projekt folyamat mely fázisában kell elkészülnie?
  Az IT rendszer implementációt a minden IT projektben jelen levő implementációs szakaszban végezzük el. Célja az adott szoftver terméket futtató IT infrastruktúra előállítása.

- Mely projekt szereplők feladata ezeket elkészíteni?
  Az IT infrastruktúrát számos IT szervezet valósítja meg: hálózat üzemeltetés, gépterem üzemeltetés, szerver adminisztrátorok, adatbázis adminisztrátorok, alkalmazás szerver adminisztrátorok, alkalmazás adminisztrátorok.

- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
  A szakmai termék fogalma itt maga az IT infrastruktúra, ami installálásra és konfigurálásra kerül. Kivitelezési tervek esetleg készülhetnek, amelyek a tervezési dokumentumokra alapozva:

  Technológiai tervezés:
  - Deployment diagramok
  - Fizikai szerver, adatbázis, alkalmazás szerver architektúra
  - Hálózati architektúra
  - Logikai és fizikai kapcsolati mátrix
  - Hardver és szoftver méretezés
  - Infrastruktúra rendszer biztonsági dokumentáció
  - Katasztrófa terv (DRP) terv
  - Infrastruktúra mentési terv
Infrastruktúra felügyeleti terv

- a részletes implementációs tervet tartalmazzák, de ez nem minden esetben szükséges, a projekt komplexitásától függ.

- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?

Az IT infrastruktúra megtérülési mutatója lényeges eleme az adott IT rendszer megtérülésének és a költség szintjének is lényeges eleme, de részletes elemzése nem kapcsolódik szorosan a dolgozatom témájához, ezért nem elemzem.

Azonban annyit mindenképpen érdemes megemlíteni, hogy az egyik legkevészbé rugalmasan változtatható szakmai termék egy IT projektben, az infrastruktúra beszerzés és építés, konfiguráció egy lassú és rugalmatlan folyamat, szinte kizárólag vízesés elven szoktuk végezni a tervezését és implementációját.

Ez az agilis fejlesztési projektek leggyengébb pontja is: az IT architektúra nem változtatható sprintről sprintre, a terveket a projekt elején véglegesíteni kell és utána már csak nagyon erőforrás igényesen és nagy átfutási idővel változtathatóak.

Az elmúlt évek felhő alapú informatikai fejlesztései lehetővé teszik hogy az eddigi modellel szakítva lényegesen nagyobb rugalmasságot, drasztikusan lerövidült implementációs időket és alacsonyabb költségszinteket érjünk el, ezért van remény arra hogy a jövőben ez változni, javulni fog, de jelenleg még a folyamat elején tartunk csak.
Tesztelés szakasz

A PRINCE2 módszertan a minőség biztosítására két fő fázist határoz meg, ahogy az ábrán is látható:

- Minőség tervezés
- Minőség ellenőrzés

A szoftver minőségét tervezni többek között tesztkészítésével tudjuk, a szoftver minőségét ellenőrizni pedig teszteléssel tudjuk.
A tesztelés mottója:

„A hibátlan rendszer téveszméje: Hiába javítjuk ki a hibákat a szoftverben, azzal nem lesz elégedett a megrendelő, ha nem felel meg az igényeinek. Azaz használhatatlan szoftvert nem érdemes teszteni.”

(Szoftvertesztelés, Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán, 2009)

- A projekt folyamat mely fázisában kell elkészülnie?

A projekt tesztelési szakaszának célja, hogy a szakmai termékekben esetlegesen meglevő hibákat megtaláljuk és megszüntessük, hogy növeljük a megbízhatóságot és minőséget. Ezt teljes körűen nem lehet megtenni, korlátot szab a rendelkezésre álló idő és erőforrás. A tesztelés a teszt tervek alapján kerül végrehajtásra, ami a rendszertervezési szakaszban elkészült funkcionális és technológiai teszt tervekre illetve konkrét teszt esetekre épül. A tesztelési fázisban a tesztek végrehajtását teszt jegyzőkönyvekben rögzítjük, melyek alapján kiértékeljük az eredményeket és összevetjük a tesztelési fázis kilépési feltételeivel (ami a teszt tervben lett rögzítve, mint sikerességi kritérium). A tesztelés végén jelentős készül, ami az alapja a következő projekt szakaszba való átlépésnek.
A tesztelést érdemes a projekt minél korábbi szakaszában elkezdeni, mert minél hamarabb sikerül megtalálni egy hibát, annál alacsonyabb költséggel lehet javítani, ahogy az ábrán is látható:

18. ábra, Hibajavítás költséggörbe (Molnár Bálint, A projekt irányítás módszertena, 1997)

A tesztelés kockázat kezelési kérdés is: mivel a tesztelésre csak véges erőforrásunk van, érdemes a teszteket a leggyakrabban használt, legkritikusabb funkciókra korlátozni, vagy azokra amelyekben legvalószínűbben fordulnak elő hibák.
• Mely projekt szereplők feladata ezeket elkészíteni?
• A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
  • A PRINCE2 módszertan szerint a minőség ellenőrzés a V-modell szerint épül fel, ami a szoftvertesztelésre vonatkoztatva a következő teszt típusokat jelenti:

![V-Modell diagram](image)

19. ábra, V-Modell (Szoftvertesztelés, Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán, 2009)

• A komponensteszt (Unit teszt, Modul teszt) a szoftver egyes önállóan működő részeit teszteli. Jellemzően még a fejlesztési fázisban elkezdődik ez a tevékenység és a rendszer önálló részeit teszteli általában a forráskód ismeretében, a fejlesztő részvételével. A unit teszt és modul teszt közötti lényeges különbség, hogy a unit teszt
a kódrészlet funkcionális működőképességét vizsgálja, míg a modulteszt a nem funkcionális követelményekre koncentrál (performancia, hibakezelés).

- Az integrációs teszt a rendszer egyes részei, vagy több rendszer egymás közötti együttműködés helyességét vizsgálja. Az integrációs tesztelést is érdemes már az implementációs fázisban elkezdeni, mert minél komplexebb az integrált rendszer, annál nagyobb a valószínűsége, hogy a rendszerek közötti együttműködés nem lesz problémamentes. Az integrációs szintaktikai problémákat a tervezés során interfész szerződésesek specifikációjával lehet csökkenteni, illetve minél korábbi interfész tesztekkel. A funkcionális, szemantikai együttműködési problémák felfedezéséhez pedig funkcionális végponttól végpontig tartó folyamati teszteket kell futtatni, amint lehet.

- Amennyiben a komponens vagy integrációs tesz eseteket a változtatások között egymás után többször futtatjuk, azt regressziós tesztnek nevezzük, amin arról próbálunk meggyőződni, hogy a változtatások nem okoztak hibát.

- A rendszerteszt a projekt teljes IT szakmai termékét teszteli, hogy megfelel-e a követelmény meghatározási fázisban lefektetett követelmény specifikációznak, funkcionális és nem funkcionális terveknél illetve a rendszertervnek. A rendszertesztet jellemzően még az IT szakmai szervezetek és a projekt szakmai teszt csapata végzi, nem része az átvételi teszteknek.

- Az átvételi teszt, vagy gyakran felhasználói átvételi teszt (User Acceptance Test –UAT) célja a rendszer funkcionális elfogadása a megrendelő részéről. Ideális esetben ez alatt a tesztek alatt már nem derülnek ki hibák, ha mégis, ebben a fázisban már csak magas költséggel lehet javítani őket, mert az összes előző teszt körön végig kell menni még egyszer a javított funkcióknak.

- Az üzemeltetői átvételi teszt (Operational Readiness Test –ORT) célja a felhasználói átvételi teszthez hasonlóan a rendszer elfogadása a megrendelő részéről, de funkcionális szempontok helyett a nem
funkcionális követelményeknek való megfelelőséget vizsgálja. Része a funkcionális specifikációban illetve a technikai specifikációkban szereplő nem funkcionális követelményeknek való megfelelés ellenőrzése (hibakezelés, rendelkezésre állás..)

- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?
  A tesztelés rendkívül erőforrás igényes tevékenység, a tesztelési erőforrás ráfordítás összemérhető az implementációs szakasz erőforrás igényével. Az erőforrás igényt kétféleképpen lehet csökkenteni:
  - jobb minőségű termékkel, így a tesztelési ciklusok száma és az újratesztelések száma csökken
  - új típusú teszt eszközökkel, technikákkal, pl. teszt automatizálás

A teszt típusok automatizálhatósága eltérő, ahogy az ábrán is látszik:

![The Agile Testing Quadrants](image)

Az implementációs fázisban a komponens tesztek rendkívül jól automatizálhatóak, a fejlesztő eszközök pedig széleskörű támogatást adnak ehhez.

Az integrációs és rendszer tesztek automatizáltsági foka is lehet magas, azonban ebben az esetben már több erőforrást kell befektetni a folyamati tesztek automatizálásába, de a legtöbb esetben ez visszatérül.

Az üzleti felhasználók számára tartott bemutató teszek, a felhasználói átvételi tesztek nem automatizálhatóak, mert itt a cél a tényleges humán felhasználás bemutatása és tesztelése.

Az üzemeltetői átvételi tesztek egy része informatikai eszközökkel könnyen automatizálható: betörési és biztonsági tesztek, illetve az integrációs, folyamati tesztek tömeges futtatásával a teljesítmény és terheléses tesztek. Más részük, például a rendelkezésre állási, katasztrófa tűrési tesztek erőforrás igényesek, mert jellemzően egyszer hajtjuk végre őket ezért nem érdemes automatizálni.

A projekt zárás után a fent említett tesztek esetekre továbbra is szükség van, mert akár funkcionálisan fejlesztjük tovább egy későbbi projektben a rendszert ezek újra felhasználhatóak, mint regressziós tesztesetek, akár az IT változáskezelés nem funkcionális területein történik a továbbfejlesztés: infrastruktúra változások, szoftver és hardver frissítések során, szükséges lehet a rendszer regressziós újra tesztelése. Ezért az automatizált tesztesetek tervezésébe és fejlesztésébe fektetett erőforrás és költség megtérülése magas lehet, sokszorosan megtérülhet a termék teljes életciklusa alatt.

Az egyre gyorsabban változó vállalati környezetben felértékelődik annak a szerepe, hogy a bizonytalanságot a célokban, követelményekben és az implementációs, tesztelési szakaszban hogyan tudjuk kezelni, ennek megfelelően a változás kezelés költsége egyre nagyobb hangsúlyt kap. Emellett a projektek megvalósítási költsége is főkuszba került, ami kikényszerítette a hagyományos vízesés modell mellett újabb módszertanok megjelenését, amelyek jobban megfelelnek a megváltozott követelményeknek.
Röviden bemutatom ezeket a módszertanokat a teljesség igénye nélkül, a tesztelési vonatkozásaikra koncentrálva:

- Az iteratív módszertanok szerint a projekt szakaszait részekre bontjuk és több iterációval végezzük el. Egy cikluson belül lehet elemzés, tervezés, implementáció, tesztelés és akár felhasználói átadás/átvétel is. Az iteráció célja a folyamatos finomítás, a minőség és részletezettség folyamatos javítása. A tesztelés folyamatos, első sorban az új és megváltozott funkciókra vonatkozik, de regressziós tesztelest is tartalmazhat. Ezt a módszertant alkalmazzák a RAD (Rapid Application Development) és az agilis fejlesztési modellek is. A fő előnye ennek a módszertannak a párhuzamosítás, az egyes lépések időben átfedhetik egymást. A tesztelés pedig kifejezetten jól párhuzamosítható tevékenység.

21. ábra, Iteratív módszertan (Szoftvertesztelés, Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán, 2009)

- Az agilis fejlesztési módszertan az iteratív módszertan egy alesete. Pontos fejlesztési módszertani definíciója nincs (meglátásom szerint szándékosan, ez a módszertan része), de egy Agile Manifesto kiadványban összefoglalták a fő alapelveket:
  - személyes kommunikáció előtérbe helyezése -> szabályzatok, folyamatok használatának háttérbe szorítása
  - működő szoftver a cél -> nem a dokumentáció és egyéb szakmai termékek
  - megrendelővel történő együttműködés -> a részletes szerződés és specifikáció helyett
rugalmas változáskövetés.

Példa az agilis szoftverfejlesztési módszertan implementációjára (Scrum):

A hagyományos vízesés alapú és agilis projektekben a tesztelést összehasonlítva a következőt állapíthatjuk meg:

- Követelmény kezelés: Hagyományosan a tesztelés megkezdése előtt a követelmény felmérés fázis befejeződik. Az agilis módszertanok ciklikus munkamódszere miatt a követelmények folyamatosan fejlődnek.
- Teszt automatizálás: A tesztek nagy része manuális a hagyományos projektekben, mert az újrafelhasználhatóságuk a projekt időszak alatt korlátozott, a teljes életciklus figyelembe vétele pedig nem a módszertan része. Agilis projekt esetén az újrafelhasználhatóság a projekt időszakban lényegesen nagyobb, ezért az automatizálás foka magas lehet.
- Ki tesztel? Hagyományosan a tesztelők, akiknek ez a fő feladatuk. Agilis módszertan esetén a teljes csapat, beleértve követelmény kezelőket és fejlesztőket is.
Az agilis szoftverfejlesztés első olvasásra egy nagyon hatékony csapatmunkán alapuló, de kevésé formális projektet feltételez. Ezt nagyon könnyű félreértelmezni és könnyen káoszba, követhetetlen és számon kérhetetlen projektbe alakulhat át idővel az amit hatékonyságnak és agilisnak nevezünk.

A legfontosabb tanulságai az elmúlt évek agilis projekt tapasztalatainak:

- A hatékony ciklusok, sprintek alapja, hogy minden előfeltétel és függőség rendelkezésre álljon és a projekt folyamata pontosan és jól működjene, hogy a valódi célra tudjon koncentrálni a csapat.
- A projekt szakmai termékeinek azt részét nem szabad agilis módon készíteni, aminek a változáskezelése vagy időben, vagy erőforrásban aránytalanul nagy (például technikai architektúra) ezért egy sprintbe nem fér bele.
- A hibázás a módszertan része (ez az egyik legnehezebb kulturális váltság). Ez azt jelenti, hogy nem szabad időt és erőforrást a hibajavításokra szánni egy cikluson belül. Ha nem működik, meg kell buktatni és új ciklust kezdeni.
- Az agilis nem csak az emberek hozzáállásán múlik, hanem eszközökön és munkamódszereken is, tanulni kell.

Az agilis projekt tapasztalatok utolsó, eszközöket és munkamódszereket hangsúlyozó ponthoz kapcsolódva a tesztelési vonatkozását az agilis módszertanoknak egy példán keresztül kiemelve mutatnám be:

Az extrém programozás egy olyan agilis módszertan, ami a munkamódszerekre és technikákról koncentrál és ezek minél jobb alkalmazására, a minél magasabb minőség elérésének az érdekében. A tesztelés kapcsán a fő koncepciója, hogy egy funkciót akkor tudunk elfogadni működőnek, ha teszteltük. Ezért amennyit csak lehet annyit kell teszteni, mert a tesztelési tevékenység többszörösen visszatérül:

- Minél többet tesztelünk annál több hibát találunk meg, ami egyenesen arányos a minőséggel
- A tesztelés helyettesíti a dokumentációt: a unit tesztek specifikációjá önmagában dokumentálja a metódusokat
- Követelmény specifikáció helyett átvételi teszt eseteket készítünk és ezt fogadtatjuk el a megrendelővel
A fenti pontokból összefoglalásából következik és fejlődött ki a Teszt vezérelt fejlesztés (TDD: Test Driven Development) módszertan: a fejlesztés megkezdése előtt elkészítjük az adott modulhoz tartozó teszteseteket és akkor tekintjük a fejlesztés befejezettnek, ha a teszt esetek sikeresek. A módszertan leírása az ábrán látható:

A minél több tesztelést segíti el a Folyamatos integráció (CI: Continous Integration) alkalmazása, ezt az jelenti, hogy a kódokat folyamatosan (pl. nap végén) telepítjük és teszteljük, jó esetben a teljes integrációs tesztet lefuttatva rájuk. Így a lehető legkorábban megtalálhatjuk az együttműködési problémák.

Az utolsó de egyik legfontosabb technika a teszt automatizálás, ami elősegíti a minél több tesztelést. Teszt robotok alkalmazásával és a teszt esetek ezekben való implementálásával jelentős idő és erőforrás szükségletet takaríthatunk meg.

Összefoglalva a tesztelés az egyik leg erőforrás igényesebb része az IT projekteknek. A tesztelés IT szakmai termékeinek megtérülése a hagyományos, PRINCE2 által leírt projekt menedzsment módszertanok használata esetén a projekt időszak alatt korlátos. A teljes életciklusra vetítve már magasabb lehet. Az iteratív, agilis módszertanok használata esetén a tesztelési szakmai termékeinek az újra felhasználhatósága már lényegesen magasabb, az iterációk alatt többszörös a megtérülésük. Ezért feltételezhetjük, hogy ezen módszertanok térhódítása elő fogja segíteni azt, hogy a projektek tesztelés kapcsán is olyan szakmai termékeket állítsanak elő, amelyek a projekt zárás után is a termék teljes életciklusa alatt újra felhasználhatóak, például automatizált regressziós tesztelésre.

23. ábra, Tesztvezérelt fejlesztés (Szoftvertesztelés, Ficsor Lajos, Kovács László, Kusper Gábor, Krízsán Zoltán, 2009)
Oktatás szakasz

A PRINCE2 projekt menedzsment módszertan nem ad részletes útmutatót az oktatással kapcsolatban. Két helyen említ, a bevezetőben felhívja a figyelmet a projekt résztvevők projekt menedzsment képességekre való oktatásának szükségességére, illetve az előre haladás kezelése fejezetben. A példa ábra alapján:

![PRINCE2 szakasz menedzsment diagrama](image)

**24. ábra, PRINCE2 szakasz menedzsment (OGC, Managing Successful Projects with PRINCE2, 5th ed)**

- A projekt folyamat mely fázisában kell elkészülnie?

a PRINCE2 megalkotói szerint az oktatás tervezése a tervezési szakaszban kezdődik, az oktatási anyagok megalkotása pedig az implementációs fázis végéig tart.

- Mely projekt szereplők feladata ezeket elkészíteni?

Az oktatási anyagok elkészítése nagyon gyakran külön projekt csapat feladata. A funkcionális követelmények és funkcionális tervek az alapja a felhasználói oktatásnak, ebből készül az implementációs fázis során a konkrét rendszerre vonatkozó oktatási anyag. Ezért a projektben résztvevő üzleti/megrendelői képviselők és funkcionális tervezők közös munkája szükséges a szakmai anyagok elkészítéséhez. A rendszer specifikus, technológiai oktatások a technikai tervek alapján készülnek, mélységük és mennyiségük erősen függ a megrendelő szakmai kompetenciájától.
A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak? A funkcionális oktatási szakmai anyagok több szervezet számára is felhasználhatóak:

- Üzleti felhasználók – használat
- HelpDesk – Incidens kezelés
- Alkalmazás támogatás – Incidens és probléma kezelés
- Tesztelők

ezért az elkészítésükre fordított erőforrás többszörösen megtérülhet.
A technikai oktatási anyagok általában egy adott szakmai szervezet számára szólnak, szerepük fontos de újra felhasználhatóságuk korlátozott.

Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?
Az oktatási szakmai anyagok elkészítésének az elsődleges célja az üzleti és szakmai felhasználók számára a szükséges ismeretek átadása a rendszer használatára vételéhez. Ez a tevékenység azonban a projekt zárás után is folytatódik: a szervezetben a felhasználók cserélődnek, a termék életciklusa alatt az oktatási anyagokra folyamatosan szükség van. Ezért mindenféle jó a megtérülése az ezek elkészítésébe fektetett erőforrásoknak. A funkcionális oktatási termékeknek magasabb a megtérülése a szélesebb felhasználói körnek megfelelően.
Az elmúlt évek költségcsökkentési elvárásai kikényszerítettek hogy hatékonyabb oktatási módszereket alkalmazzunk a projekteken belül. Ezekre példák:

- Train the trainer: nem oktatjuk le közvetlenül a teljes felhasználói bázist. Oktatókat képzünk ki, akik ugyancsak oktatókat képeznek ki. Így az oktatás hatékonyabb és átfutási ideje lényegesen rövidebb lehet.
- e-Learning: a hagyományos tantermi oktatások helyett elektronikus oktatási anyagok készülnek, amelyeket szabadon választott
helyszínen és időben meg lehet tekinteni. Így az oktatásra fordított idő lényegesen kevesebb része mehet el a munkára fordított időből.

- Oktatási szimulációs rendszerek: az oktatásokat nem az integrált teszt rendszeren (az oktatás ideje a legtöbb esetben egybeesik a teszteléssel ezért nehéz ugyanazon a rendszeren időosztásban végezni ezt) és nem is integrált oktatási rendszeren (aminék a megépítése és fenntartási költsége magas lehet), hanem egy interfész szimulátorokkal körbevett szigetrendszeren tartjuk. Ennek az előnye, hogy nem kell a teljes üzleti folyamatokat és integrációt megvalósító oktatási infrastruktúrát kiépíteni, mégis a felhasználók számára valódi rendszeren történhet az oktatás. Az interfész szimulátorok nagy része a fejlesztési fázisban el tud készülni, mert a fejlesztési tevékenységet is tudja támogatni a meglétük (integrációs teszt előtt lehet simulálni az interfészeket így a unit tesztek jobb minőségűek lesznek), többszörösen visszatérülhet a kifejlesztésük.

- Agilis team: az agilis módszertanok része, hogy minden team résztvevő tesztel, ami feltételezi a megfelelő ismereteket is, amit oktatással lehet megszerezni. Ezért a tesztelés és oktatás agilis projekt esetén egy időben történik, nem külön tevékenység, így lényegesen hatékonyabb tud lenni. A formális oktatási anyagok helyet a teszt tervek szolgálnak információ forrásul.
Projet zárás

A projekt zárást a PRINCE2 módszertan részletesen definiálja, a fő fázisok az ábrán láthatóak:

- A projekt folyamat mely fázisában kell elkészülnie? A szakmai termékek átadását a megrendelő részére, az üzembe állítást és projekt zárást a PRINCE2 módszertan szerint a projekt zárási szakaszban kell megtenni. A projekt zárás célja a megrendelői elfogadás, a kitűzött célok elérésének a kiértékelése és a projekt szervezet megszüntetése.
- Mely projekt szereplők feladata ezeket elkészíteni? A projekt nem szakmai termékeinek a formális zárását (Projekt indítási dokumentáció, Projekt terv, Issue Register, Risk register, Quality Register, Lessons log..) a projektvezető és a projekt szervezetben az adott termékért
felelős szereplők végzik. Részletesen nem elemzem ezt a tevékenységet, mert a dolgozatomnak nem ez a témája.

- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
  A projekt szakmai termékeinek az zárásához és átadás/átvételéhez a következő szakmai termékek kapcsolódnak, amelyeket ebben a fázisban kell elkészíteni vagy véglegesíteni:
    - Szolgáltatás felelős és menedzser (Szolgáltatás felelős: aki az adott IT szolgáltatásért felel a vállalaton belül minden tekintetben, Szolgáltatás menedzser: aki az adott IT szolgáltatás napi működéséért felel): Ha a projekt indítási fázisban vagy a későbbi szakaszokban nem került sor a kijelölése, a projekt zárás részeként mindenképp meg kell tenni, hogy a megrendelő számára át lehessen adni a projekt által készített IT termékkel kapcsolatos felelősséget.
    - ITIL folyamatok: A projekt zárási tevékenységek közé tartozik az ITIL által definiált folyamatok (Incidens kezelés, Probléma kezelés, Változáskezelés..) üzembe állítása. Ehhez mindenképpen szükséges:
      - az üzemeltető szervezetek definiálása
      - a szállítói és megrendelői szervezetek feladat elhatárolásának rögzítése a támogatási szerződések alapján
      - a folyamatok működtetéséhez szükséges eszközök feltöltése a projekt által szállított IT termék adataival (IT és üzleti szolgáltatások definíciója, konfigurációs elem adatbázis aktualizálása(CMDB))
    - Produkcióba kirakási terv
      A produkcióba kirakási terv az élesítési (Golive) terv részletes IT nézete, ami tartalmazza az adott IT termék üzembe állításához szükséges adminisztrátori és operátori teendőit, órákra/percekre lebontva. Ez a terv szükséges ahhoz, hogy a lehető legrövidebb üzleti állásidővel, jó minőségben történjen meg a produkcióba állás.
    - Üzleti funkciók élesítési terve
      Az üzleti funkciók élesítési terve akkor releváns egy projektben, ha az IT rendszer üzembe állítása időben nem esik egybe annak üzleti
célú használatának a megkezdésével. Ebben az esetben külön terv szükséges az üzleti funkciók élesesítéséhez.

- Pénzügyi zárás: A projekt pénzügyi szemléletű zárásához a forrásokkal való elszámolás és a capex/opex forrásokhoz való hozzáférés zárás mellett a következő tevékenységek kapcsolódnak, amelyek egy része IT szakmai tartalmat is hordoz: A IT tárgyi eszközök aktiválási bizonylatainak elkészítése, szállítói teljesítés igazolási bizonylatok kiállítása, IT átadás/átvételi jegyzőkönyvek elkészítése.

- SLA szerződés megkötése: Amennyiben az adott projektben az üzleti megrendelőnek igénye van erre, az adott megrendelői IT szakmai szervezettel vagy beszállítóval szolgáltatási szint szerződést (SLA: Service Level Agreement) köthet a projekt zárási szakaszában. Szerencsés esetben a tervezett SLA feltételei a projekt indítás óra ismertek és a rendszer tervei ennek megfelelően készültek el.

- Támogatási és üzemeltetési szerződések megkötése: A szállító kiválasztás szakaszban már részletesen ismertettem a támogatási és üzemeltetési szerződések megkötésének szempontjait, ezért itt nem ismételném meg újra, csak megemlítem, hogy a szerződés megkötésének a javasolt időpontja a projekt zárás szakasa.

- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?
  A projekt zárási szakaszban elkészülő IT szakmai termékek jelentős része egyszeri célt szolgál, az újra felhasználhatósága és projekt zárás utáni megtérülési mutatója alacsony. Azonban van két projekt termék amibe a befektetett erőforrások a projekt zárás után is jelentős megtérülést hoznak:
  - Lessons Log: A PRINCE2 módszertan szerint érdemes a projekt zárási szakaszban egy Lessons review-t (tanulságok összefoglalása) tartani a projekt szakaszok alatt rögzített tanulságok alapján. Ennek mindenképpen lehet IT szakmai tartalma is, amit érdemes az adott szállítói és megrendelői tudásmenedzsment eljárás szerint megosztani
és az alkalmazott szabályozásokba, módszertanokba, tudásbázisokba
visszavezetni, ennek a megtérülési mutatója nagyon magas lehet.

- A befektetés indokolás: A dolgozatom elején részletesen elemeztem a
befektetés indokolást, ezért itt csak röviden utalnék arra, hogy a befektetés
indokolás aktualizálása a projekt zárási szakasznak fontos feladata. A
PRINCE2 módszertan is külön kiemeli a fontosságát azért, mert tartalmazhat
olyan teendőket, amelyeket a projekt zárás után kell elvégezni, hogy a
megtérülés biztosítva legyen.
Összefoglalás

Az IT projektek szakmai termékeinek megtérülését elemeztem a dolgozatomban a PRINCE2 módszertan alapján. A projekt szakaszokon keresztül felvetettem azokat a kérdéseket, amelyek alapján választ lehet arra találni, hogyan lehet a szakmai termékeket minél jobb, a teljes termék életciklusra vetített megtérüléssel (TCO) elkészíteni:

- A projekt folyamat mely fázisában kell elkészülnie?
- Mely projekt szereplők feladata ezeket elkészíteni?
- A szakmai termékek milyen szervezetekhez, folyamatokhoz kapcsolódnak?
- Milyen szempontokat kell figyelembe venni a minél jobb megtérülés érdekében?

Amelyik szakaszban értelmezhető volt, felvetettem azt a kérdést is, hogy a dolgozatomban elemzett, projektzárás után is felhasznált szakmai termékeket hogyan, milyen szakmai módszertanok alkalmazásával érdemes elkészíteni? Mennyire kell ragaszkodni a hagyományos PRINCE2 módszertanhoz, ami arra koncentrál, hogy a projekt fázisok sorosan egymásra épülnek és a termék használatba vételének időpontjára minden projekt szakmai termékek végleges formában el kell készülnie, vagy érdemes az új módszertanok által képviselt iteratív, inkrementális vagy agilis módszertant követni?

Fontos felismerés, hogy az új, iteratív és agilis módszertanok ugyanazt a szakmai terméket többször is felhasználják már a projekt időszak alatt is. Így a termékek elkészítéséhez és változáskezeléséhez használt szakmai módszerek, folyamatok fejlődése lehetővé teszi, hogy ezek a szakmai termékek ne csak azt a projekt egyszeri célt szolgálják, hogy a vízesés módszertan szerint a következő fázis alapjai legyenek. Az újra felhasználás és a folyamatos változáskezelés az projekt módszertanok része, ezért erre alapozva ez a tevékenység a projekt időszak alatt és a projekt zárása után is ugyanolyan hatékonysággal folytatódhat, ha ez a projekt célok között megjelenik. Ugyancsak fontos felismerés, hogy a projekt szakmai termékeinek folyamatos
változtatási, finomítási lehetősége tervezett, és a tervezési szakaszban már felismert tevékenység legyen. Lehet, hogy már a projektzárás után történik az újra felhasználás és változáskészlels, de a költsége meg legyen jelenítve a befektetés indokolási tervben, illetve a megfelelő szakmai felelősök, szervezetek ki legyenek jelölve a projekt zárás utáni feladatok elvégzésére már a projekt indítási szakaszban, hogy a követelmény kezelés részei lehessenek. Természetesen ez minden szakmai termékre nem alkalmazhatóak ugyanolyan hatékonysággal a fentiek, de például a felhasználói kézikönyv, tudásbázis, regressziós tesztesetek, oktatási anyagok.. ilyenek lehetnek.

Fontos, hogy ezek az alapelvek a projekt célok között is megjelenített, kiemelt célok legyenek, hogy ezek az IT szakmai és költséghatékonysági célok a megrendelő számára is világos és elfogadott célokká tudjanak válni.
<table>
<thead>
<tr>
<th>Ábrajevgyezék</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ábra: IT termék életciklus (Ficsor Lajos, Dr. Kovács László, Krizsán Zoltán, Dr. Kusper Gábor – 2009, Szoftvertesztelés) .......................................................... 17</td>
</tr>
<tr>
<td>2. ábra, Prince2 összetevői, (BCE GTK Információrendszerek Tanszék (2IR32NDV01M_GINF1_2_G1 kurzus) .......................................................... 18</td>
</tr>
<tr>
<td>3. ábra, Business Case (OGC, Managing Successful Projects with PRINCE2, 5th ed) .......................................................... 22</td>
</tr>
<tr>
<td>5. ábra, Projekt szervezet (OGC, Managing Successful Projects with PRINCE2, 5th ed) .......................................................... 24</td>
</tr>
<tr>
<td>6. ábra, Projekt szervezeti struktúra, (OGC, Managing Successful Projects with PRINCE2, 5th ed) .......................................................... 30</td>
</tr>
<tr>
<td>7. ábra, ITIL folyamatok (ITIL - az informatikaszolgáltatás módszertana, 2002 - KFKI Számítástechnikai Rt.) .......................................................................................................................... 31</td>
</tr>
<tr>
<td>8. ábra, Projekt szervezeti szerepkörök (Agile Change Management Limited, 2013, Comparing the roles from PRINCE2 with agile project management) ................................................................ 32</td>
</tr>
<tr>
<td>9. ábra, PRINCE2 projekt terv (OGC, Managing Successful Projects with PRINCE2, 5th ed) .......................................................... 34</td>
</tr>
<tr>
<td>10. ábra, Gant chart példa .......................................................................................................................... 34</td>
</tr>
<tr>
<td>11. ábra, PRINCE2 projekt szakasz kontroll (OGC, Managing Successful Projects with PRINCE2, 5th ed) .......................................................................................................................... 36</td>
</tr>
<tr>
<td>12. ábra, Vízéses modell (AGILE - SCRUM Project Management, 2007, Project Lifecycle Services Ltd) .......................................................................................................................... 37</td>
</tr>
<tr>
<td>13. ábra, Agilis módszertan ciklus (AGILE - SCRUM Project Management, 2007, Project Lifecycle Services Ltd) .......................................................................................................................... 38</td>
</tr>
<tr>
<td>15. ábra, UML diagrammok (Szoftverfejlesztés, Ficsor Lajos, Kovács Zoltán, Dr. Mileff Péter, 2011) .......................................................................................................................... 42</td>
</tr>
<tr>
<td>16. ábra, Hibajavítás költség gördbe (Molnár Bálint, A projekt irányítás módszertana, 1997) ................................................................ 46</td>
</tr>
<tr>
<td>17. ábra, PRINCE2 minőségbiztosítás (OGC, Managing Successful Projects with PRINCE2, 5th ed) .......................................................................................................................... 52</td>
</tr>
<tr>
<td>18. ábra, Hibajavítás költség gördbe (Molnár Bálint, A projekt irányítás módszertana, 1997) ................................................................ 54</td>
</tr>
<tr>
<td>19. ábra, V-Modell (Szoftvertesztelés, Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán, 2009) .......................................................................................................................... 55</td>
</tr>
<tr>
<td>21. ábra, Iteratív módszertan (Szoftvertesztelés, Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán, 2009) .......................................................................................................................... 59</td>
</tr>
<tr>
<td>22. ábra, SCRUM módszertan (Szoftvertesztelés, Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán, 2009) .......................................................................................................................... 60</td>
</tr>
<tr>
<td>23. ábra, Tesztvezérelt fejlesztés (Szoftvertesztelés, Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán, 2009) .......................................................................................................................... 62</td>
</tr>
<tr>
<td>24. ábra, PRINCE2 szakasz menedzsment (OGC, Managing Successful Projects with PRINCE2, 5th ed) .......................................................................................................................... 63</td>
</tr>
<tr>
<td>25. ábra, PRINCE2 projekt zárás (OGC, Managing Successful Projects with PRINCE2, 5th ed) .......................................................................................................................... 66</td>
</tr>
</tbody>
</table>
Szakírodalom lista


- Broczkó Péter (2011), ITIL alapú szolgáltatás menedzsment, Typotex Kiadó

- Molnár Bálint, A projektirányítás módszertana, 1997, MTA ITA


- Klimkó Gábor, 2IR32NDV01M_GINF1_2_G1 kurzus prezentáció , BCE GTK Információrendszerek Tanszék, 2011


Internetes források jegyzéke

- IT Projektek megtérülése, Dr. Fehér Péter, Budapesti Corvinus Egyetem, Információrendszerk tanszék
  Letöltés ideje: 2014.05.01, Letöltés helye:
  http://www.slideshare.net/pethich/it-projektek-megtrelse

- Ficsor Lajos, Kovács László, Kusper Gábor, Krizsán Zoltán, 2009, Szoftvertesztelés, Kelet-Magyarországi Informatika Tananyag Tárház
  Letöltés ideje: 2014.05.01, Letöltés helye:

- The Business Case for Agile, Seattle Scrum Gathering, 2011 SolutionsIQ.
  Letöltés ideje: 2014.05.01, Letöltés helye:
  http://www.scrumalliance.org/resource_download/2336

- ITIL - az informatikaszolgáltatás módszertana, KFKI Számítástechnikai Rt, 2002
  Letöltés ideje: 2014.05.01, Letöltés helye:

- Comparing the roles from PRINCE2® with Agile Project Management, Agile change management limited, 2013
  Letöltés ideje: 2014.05.01, Letöltés helye:
- AGILE - SCRUM Project Management, Project Lifecycle Services Ltd, 2007
  Letöltés ideje: 2014.05.01, Letöltés helye:
  http://www.projectlifecycleservicesltd.co.uk/project-management-
  services/scrum-project-management.php

- Szoftverfejlesztés, Ficsor Lajos,Krizsán Zoltán,Dr. Mileff Péter, 2011, Miskolci
  Egyetem, Általános Informatikai Tanszék
  Letöltés ideje: 2014.05.01, Letöltés helye: