



# Szoftver fejlesztési és implementációs projektek buktatói és sikertényezői

Török Péter  
Stratis PMCC vezető

# Tematika

1

Helyzetkép

2

A szoftveripar változása

3

Kulcs a specifikáció

4

Elvárt eredmény – sikeres tesztelés

# A szoftver implementációs projektek 50-70%-a sikertelen

- **50% nem tarja az eredeti**
  - **Időtervet**
  - **Erőforráskeretet**
  - **Költségvetést**
  
- **20% megbukik**
  - **Az ügyfél nem azt kapja amit szeretett volna**
  - **Sohasem kerül bevezetésre a rendszer**

# Mindannyian a körülmények áldozatai vagyunk...

Rajtunk kívülálló  
okok

## Külső okok

- Felhasználói bevonás
- Szponzori elkötelezettség
- Követelmények meghatározása
- Prioritás
- Érdeklődés megszűnése

## Belső okok

- Erőforrás
- Szoftver specifikáció
- Rendszer specifikáció
- Szakértelem
- Elkötelezettség
- Motivációs eszköz
- Elismertség
- Munkakörülmények
- Szállítói kontrol

# Sikertényezők



**Sikeres projekt**

# Buktatók

**Idő  
terveszés**

**Erőforrás  
hiány**

**Szoftver  
specifikáció**

**Problémás  
projekt**

# Buktatók



# Tematika

1

Helyzetkép

2

A szoftveripar változása

3

Kulcs a specifikáció

4

Elvárt eredmény – sikeres tesztelés

# Gyorsan, finomat!

- Vezető beosztású ügyfél készíti
- Konceptió vagy szerződés része
- Általában nem teljes terjedelmű

Követelmény  
specifikáció

- Beosztott felhasználó készíti
- Nem a köv. spec.-ből vezetik le
- Először (utóljára) specifikál szoftvert

Szoftver  
specifikáció

- Nem jut idő a rendszertervezésre
- Specifikáció alapján kódol
- Fejlesztői tesztelés elmarad

Fejlesztés

- Felhasználó tesztel
- Először (és bizonyára utóljára)
- Kiderülnek a nem-megfelelések

Felhasználói  
átvételi teszt

- Jóval a tervezett határidő után
- Erőforrás és költségkeret túllépéssel
- Nem az amit az ügyfél (vezető) akart

Éles indulás



# Autóipar fejlődése

- ~1900 – 1950 – korai stádium
  - Csak a mérnökök értettek hozzá
  - Csak a szerelő (sofőr) tudja vezetni az autót
  - Az utas hátul ül és örül a látványnak
- 1950 – 1990 – fejlett korszak
  - A mechanikához mindenki ért
  - A szerelő karbantart és javít is
  - Az utas vezet, karbantart és néha még javít is
- 1990 – modern korszak
  - A modern elektromechanikához csak a mérnökök értenek
  - A szerelő csak karban tartani tudja az autót, ténylegesen megjavítani nem (legfeljebb alkatrészt cserél)
  - Az utas vezet, de nem nézhet a motorháztető alá

# Szoftveripar átalakulása

- ~1970 – 1990 – korai stádium
  - Csak a fejlesztők értenek a kódhoz
  - Csak az operátor tudja futtatni a programot
  - A felhasználó máshol ül, és örül az eredménynek
- 1990 – 2000 – fejlett korszak
  - Az informatikához mindenki ért
  - Az operátor karbantart és kódol
  - A felhasználó futtat, karbantart és néha kódol is
- 2000 – modern korszak
  - A modern szoftver rendszerekhez csak a fejlesztők értenek
  - Az operátor csak üzemeltetni tudja a programot
  - A felhasználó csak futtatja a programot, de nem látja a mögöttes rendszerek bonyolultságát

# Régi műsor, új szereposztásban

- Sem a fejlesztő, sem a felhasználó nem tud a másik fejével gondolkodni
- Mindenki maradjon a saját szakértői területén
- Az üzleti elemzői státuszt nem lehet megspórolni
  - Ha nincs a vállalatnál, a PM-nek kell hoznia



# Régi műsor, új szereposztásban



- Kalkulációsvesztésű kérszűri el
- Vezetői bevezetésű (ügypfű) hagyja jvív
- Teljes terjedelmű
- Üzleti elemzű / projekt szakértű kérszűri el
- Kalkulációsvesztésű hagyja jvív
- Direkt bevezetésű a követelményekből
- Fejlesztési / integrációs szerződés melléklete
- Fejlesztű kérszűri el
- Az üzleti elemzű / projekt szakértű ellenűri
- Direkt bevezetésű a specifikációból
- Kétdolok a rendszerterv szerűt
- Kétdolok, on the fly minőségbiztosítás
- Fejlesztű vagy fejlesztű minőség biztosítű végű
- Az üzleti elemzű / projekt szakértű ellenűri
- A tesztesetek a rendszertervre támaszkodnak
- Automatizálhatű
- Az üzleti elemzű / projekt szakértű végű
- A felhasználó nem tudja ellenűrizni, de a P99 igen
- A tesztesetek a specifikációra támaszkodnak
- A kalkulációsvesztésű hagyja végűre
- Az üzleti elemzű / projekt szakértű ellenűri
- A tesztesetek a követelményekre támaszkodnak

- Követelmény specifikáció
- Szoftver specifikáció
- Rendszer tervezés
- Fejlesztés
- Fejlesztű tesztek
- Funkcionális tesztelés
- Felhasználű átvételi teszt
- Éles indulás

# Tematika

1

Helyzetkép

2

A szoftveripar változása

3

Kulcs a specifikáció

4

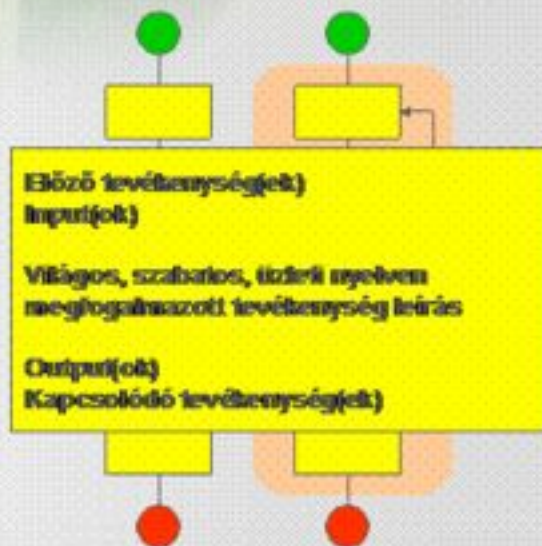
Elvárt eredmény – sikeres tesztelés

# A specifikáció a tesztelés logikáját szolgálja ki



# Követelmény specifikáció

- A tervezett új üzleti tevékenységet írja le
  - Terjedelem!
- A felhasználó üzleti nyelvén
  - A felhasználó számára legyen világos és egyértelmű
- Szabad szöveges, de szabatos leírás
- Szoftver független
  - Úgy is értelmesnek kell lennie, ha nem szoftverrel oldjuk meg a problémát
- Nem jól prioritálható



# Szoftver specifikációs dokumentumok

- **Konceptió és terjedelem**
- **Üzleti folyamatok**
- **Specifikációs dokumentum**
- **Funkciópontok prioritizálása**
- **Működési esetek (use cases)**

## SRS Tartalom

- Felhasználók jellemzése
- Üzemi környezet
- Fejlesztési és implementációs követelmények
- Dokumentációs elvárások
- Rendszer funkciók
  - ID
  - Funkciópont működésének műszaki leírása
- Belső (modul) interfész leírás
- Külső interfész leírás
  - Felhasználói (szolgáltatás)
  - Rendszer (hw, sw)
- Teljesítmény
- Biztonság
  - Adat
  - Működési
  - DRP
- Adatszótár
- Adatmodell
- Folyamat modell

# Szoftver specifikáció

- **A szoftver specifikáció a felhasználói elvárásokra adott szoftver válasz leírása**
- **A követelmények szempontjából teljes, de**
- **Több témát érint mint a funkcionalitás**
- **Műszaki (szoftver mémóki) nyelven írott**
  - Táblázatok
  - Ábrák
- **Pontokba szedett, nem szabad szöveges**
- **Sok benne a keresztihivatkozás**
  - Himl-ben könnyebb mint sima dokumentumban!

## 10 tünet, ami sikertelen specifikációra utal

- Nem világos, hogy mi a szoftver specifikáció
- Nem látta / hagyja jóvá a felhasználó (ügyfél)
- Művészi megoldások
- Funkciópontok nincsenek prioritálva
- Árva funkciópontok
- Analízis paralízis
- Követelmény -> szoftver specifikáció nem vezethető le
- Nincs változáskezelési folyamat
- A változásokat nem megfelelően elemzik
- Specifikáció verzió kontrol

# Rendszer tervezés

- Alapesetben a szállító hozza
  - A meglétét (korábbi projekt) ellenőrizni kell
- Folyamatmodellezés
  - A rendszerben zajló információs folyamatokra
  - Adatforrások és tárolás megtervezése
- Logikai adatmodellezés (ER diag.)
  - Egyed-tulajdonság-kapcsolat (Bachman)
  - Relációs modellezés (Codd)
- Fizikai reprezentáció
  - A kiválasztott fizikai rendszerplatformon történő megvalósítás módja (OS, DBMS, n-tier, MW)
- Biztonsági megfontolások
  - teljes vertikumot fedje le

# Tematika

1

Helyzetkép

2

A szoftveripar változása

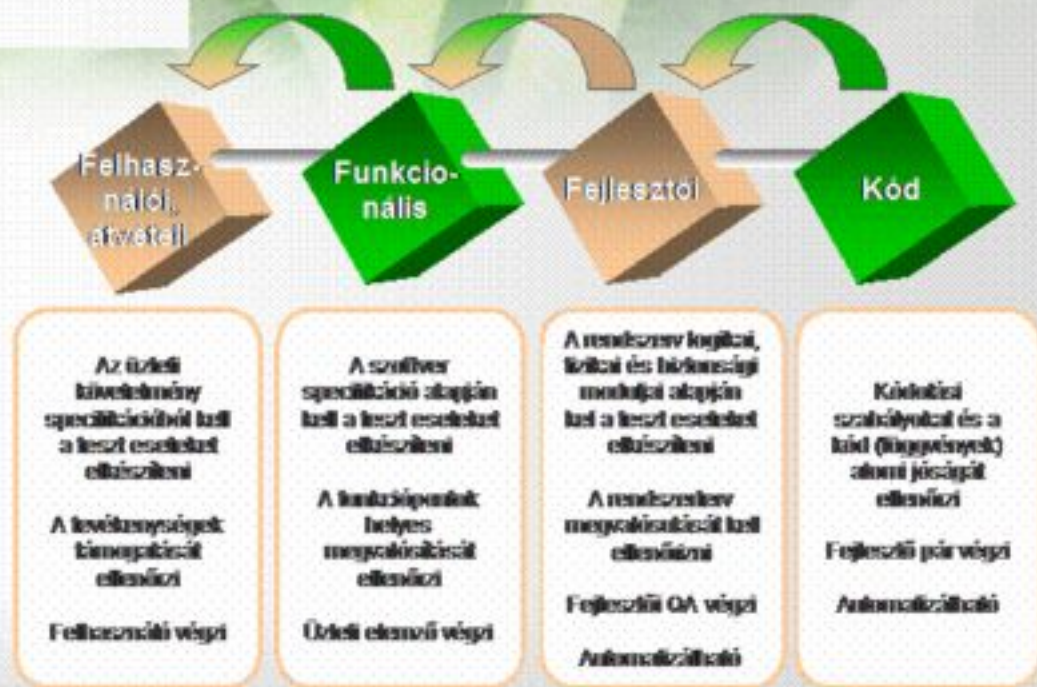
3

Kulcs a specifikáció

4

Elvárt eredmény – sikeres tesztelés

# A specifikáció a tesztelés logikáját szolgálja ki



# Tesztelési irányelvek

- A tesztelés a specifikáció megvalósulásának ellenőrzése
- Minden szinten koncepciót, tervet és teszt eseteket kell készíteni
  - A teszt csak a specifikációt ellenőrizheti
  - Nem specifikálhatunk a teszt eseteken keresztül
- A teszt esetek elkészítése összemérhető a specifikációs dokumentum elkészítésének idejével
  - Minél jobb a specifikáció minősége, annál egyszerűbb a teszt eset megcsinálni
  - A teszt tervezés és előkészítés során rengeteg specifikációs hiba derül ki
    - A változáskezelési folyamatnak a specifikáció lezárása után már működnie kell
- A tesztelés ütemezésére és ellenőrzésére használjunk teszt menedzsment támogató rendszert
  - Jó ha támogatja a hibajegy és release kezelést is

# Konklúzió

## Sikeres szoftver implementáció





**Köszönöm a figyelmet!**

<http://www.stratis.hu/>

[peter.torok@stratis.hu](mailto:peter.torok@stratis.hu)