

Dr. Mileff Péter

Szoftverfejlesztés

Gyors szoftverfejlesztés

Miskolci Egyetem
Általános Informatikai Tanszék

1

Bevezetés

- A vállalatok ma globális, gyorsan változó környezetben működnek.
 - Reagálnak az új lehetőségekre és piacokra, a gazdasági környezet változásaira.
- A szoftver része minden műveletnek,
 - Kulcsfontosságú hogy egy új szoftvert gyorsan kifejlesszenek,
 - kihasználva ezzel az új lehetőségek adta előnyöket.
- A szoftverrendszereknél így ma a legkritikusabb követelmény:
 - a gyors fejlesztés és üzembe helyezés
 - Sok vállalat hajlandó elengedni a minőségből és kompromisszumot kötni ezért.

2

Bevezetés

- Sokszor gyakorlatilag lehetetlen, hogy a szoftverrel szemben stabil elvárásokat fogalmazzunk meg.
- Nem tartoznak a gyors szoftverfejlesztéshez azok a szoftverfejlesztési folyamatok,
 - amelyek az elvárások teljes specifikációján alapulnak.
 - Az elvárások változásával szükségessé válhat a rendszerterv, illetve az implementáció átdolgozása.
- Ez gyorsan változó üzleti környezetben komoly gondokat okozhat.
 - Idővel a szoftver elkészül, de addigra a körülmények megváltozása miatt gyakorlatilag használhatatlanná válik.

3

Bevezetés

- Ez idézte elő az olyan fejlesztési folyamatokat,
 - amelyek a **gyors szoftverfejlesztésre** és átadásra összpontosítanak.
- A gyors szoftverfejlesztési folyamatokat arra tervezték,
 - hogy segítségével gyorsan készíthessünk használható szoftvereket.
- Olyan iteratív folyamat, ahol a specifikáció, a tervezés, a fejlesztés és a tesztelés átfedi egymást.

4

A gyors szoftverfejlesztés alapvető jellemzői

- 1. A specifikáció, a tervezés és az implementálás folyamata konkurens módon zajlik.
 - Nincs részletes rendszer-specifikáció
 - és a tervezési dokumentáció minimális.
 - A felhasználói elvárások leírása csak a rendszer legfontosabb jellemzőit határozzák meg.

5

A gyors szoftverfejlesztés alapvető jellemzői

- 2. A rendszert lépésről lépésre fejlesztik.
 - A végfelhasználók és a rendszer többi érintettje részt vesznek minden lépés specifikációjában.
 - Indíthatóak változások a szoftverben és új követelményeket fogalmazhatnak meg.
- 3. A rendszer felhasználói felülete gyakran egy beépített fejlesztői környezet használatával készül.
 - Ez gyors interfész elkészítést és gui elemek elrendezését eredményezi.

6

AGILIS MÓDSZEREK...

7

Agilis módszerek

- 1980-as években és az 1990-es évek elején volt egy nagyon széles körben elterjedt nézet:
 - a jobb szoftver előállításának legjobb módja:
 - gondosan megtervezük a projektet,
 - formalizáljuk a minőséggel szemben támasztott követelményeinket,
 - használjuk a CASE eszközök által biztosított elemzési és tervezési módszereket,
 - és irányított, precíz szoftverfejlesztési folyamatok használatával jutunk el a végeredményhez.

8

Agilis módszerek

- Kik szemlélet képviselői?
 - akik nagyszámú különálló programból felépülő hosszú élettartalmú szoftverrendszerek fejlesztésével foglalkoznak.
- Mi is ezzel a probléma?
 - Ha kis- és középvállalkozások rendszereinek fejlesztésekor alkalmazták, a megjelenő többletmunka olyan mértékű volt,
 - hogy gyakran ez határozta meg a szoftverfejlesztés folyamatának nagy részét
 - Több idő a tervezésre, mint a fejlesztésre és a tesztelésre.

9

Agilis módszerek

- Az időigényes megközelítéssel való elégedetlenség miatt:
 - a szoftverfejlesztők egy része az 1990-es években új, **agilis módszereket** indítványozott.
- Az indítvány célja:
 - a fejlesztőcsapat magára a szoftverre koncentráljon, ne pedig annak tervezésére és dokumentálására.
- Kiadtak egy **Agilis kiáltványt** 2001-ben.
 - 17 ember, a „*pehelysúlyú módszertanok*” néven futó módszertanok jelentős képviselői.

10

Mi is az Agilis Kiáltvány?

- A szoftverfejlesztés jobb módjait fedezzük fel azáltal, hogy csináljuk, és segítünk másoknak is csinálni.
 - Egyének és interakcióik, szemben az eljárásokkal és eszközökkel.
 - Működő szoftver, szemben a teljes körű dokumentációval.
 - Együttműködés az ügyféllel, szemben a szerződésről való alkudozással.
 - Változásokra való reagálás, szemben a terv követésével.

11

Agilis módszerek

- Az Agilis Fejlesztés egy módszertan-család,
 - nem egy konkrét megközelítése a szoftverfejlesztésnek.
 - a változás igénye idézte elő.
- Sokakban megfogalmazódott, hogy a **szoftverfejlesztés nem gyártás.**
- A cél:
 - Minél gyorsabban, minél költséghatékonyabban történjen a fejlesztés
 - az elvárt igényt minél jobban kielégítő végeredmény szülessen.

12

Az agilis fejlesztés működése

- Az **agilis szoftverképzés** egy elméleti keretrendszer.
- **Többféle agilis fejlesztési eljárás:**
 - jórészt mindegyik a kis kockázatú fejlesztésre törekszik
 - a rövid idejű fejlesztési ciklusok (ismétlések, iterációk) használatával.
- Minden ismétlés egy teljes szoftverfejlesztési ciklus:
 - Követelményelemzés, tervezés, kódolás, tesztelés és dokumentálás.
 - Egy ismétlés célja, hogy a ciklus végére egy letesztelt, elérhető kiadás jöjjön létre az alkalmazásból.

13

Az agilis fejlesztés működése

- Minden iteráció végén a fejlesztői csapat újraértékeli a projekt főbb céljait.
 - (határidők, gazdasági ügyek, feladatok)
- Az agilis módszerek a szemtől-szembeni kapcsolatot részesítik előnyben az írásos dokumentációk helyett.
 - Emiatt az agilis csoportok jellemzően egyetlen nagy irodában helyezkednek el.
- Az agilis módszereknél a haladás legfőbb mérőszáma a működő szoftver.

14

Alapelvek

1. Hasznos szoftvertermékek gyors, folyamatos szállításából fakadóan elégedett megrendelők.
2. Működő szoftver szállítása gyakran (inkább hetes, mint havi periódusban)
3. Az előrehaladás mércéje a működő szoftver
4. A követelményekben még a késői változásoknak is örülnek.
5. Szoros, napi kommunikáció a fejlesztők és a megrendelő között
6. Személyes kapcsolattartás

15

Alapelvek

7. A projekteket motivált, megbízható munkatársak vezetik
 8. Folyamatos figyelem kíséri a műszaki színvonalat és a tervet
 9. Egyszerűség
 10. Önszerveződő csapatmunka
 11. A változó körülményekhez való gyors alkalmazkodás
- Mikor célszerű használni?
- Kis projektek, kevés tapasztalt fejlesztő, gyakran változó követelmények.

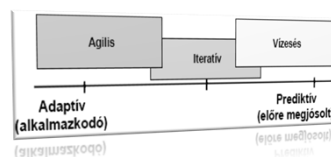
16

ÖSSZEHAISONLÍTÁS MÁ S TÍPUSÚ MÓDSZERTANOKKAL...

17

Összehasonlítás más típusú módszertanokkal

- Az Agilis módszertanokat a „terv-vezéret” vagy „fegyelmezett” módszertanok ellentétének szokták nevezni.



Arra fókuszálnak, hogy a gyakran változó követelményekhez tudjanak alkalmazkodni.

18

Összehasonlítás más típusú módszertanokkal

- Ha egy projektben megváltoznak az igények,
 - akkor egy adaptív csapat képes alkalmazkodni a változásokhoz.
- Egy adaptív csapat nehezen tudja megmondani, hogy mi fog történni a jövőben.
 - Minél távolabbi pontról van szó a jövőben, annál bizonytalanabb az elképzelés, hogy mi is fog akkor történni.
- Ha egy nagyon távoli időpontról van szó,
 - akkor a csapat már csak a vállalat célkitűzéséről, vagy a tervezett költség-érték arányról tud beszámolni.

19

Összehasonlítás más típusú módszertanokkal

- A kiszámítható módszertanok inkább arra fókuszálnak,
 - hogy minél részletesebben megtervezzék a jövőt.
- Egy kiszámítható csapat pontosan meg tudja mondani bármelyik pillanatban,
 - hogy mikor milyen feladatok lesznek készen a projektben.
- A prediktív csapatok nehezen váltanak irányt.
 - az irányváltoztatás könnyen azzal a következménnyel járhat, hogy az elkészült munkát el kell dobni, és újrakezdeni.

20

Összehasonlítás más típusú módszertanokkal

- Sokan használják a „**cowboyos kódolás nevű módszert**” is.
- Lényege:
 - Nincs semmiféle módszer meghatározva, mindenki azt csinálja, amit jónak lát.
- Sokan az agilis fejlesztést is ennek hiszik:
 - A gyakori újratervezése, szemtől-szembe kommunikációja, és viszonylag laza dokumentumkezelése miatt.
- Éles különbség van az Agilis kódolás és a cowboyos kódolás között:
 - Az Agilis csapatok nagyon is használják a maguk jól meghatározott (gyakran fegyelmezett és rigórozus) módszertanát.

21

Kritikák a módszerrel szemben

- Több kritika olvasható. Oka:
 - ezek többségükben **még nem kiforrott, lezárt módszertanok**,
 - mindenki a maga szája íze szerint értelmezheti őket.
- Kritikák három köre:
 - 1. Csak gyakorlott, szenior fejlesztőkkel működik.
 - 2. Nincs eléggé megtervezve a szoftver.
 - 3. Túl sok kulturális változás kell ahhoz, hogy jól működjön.

22

Az agilis módszerek alkalmazása

- Az agilis módszereket olyan szerződésekre kell alapozni,
 - ahol az ügyfél a rendszerfejlesztéshez szükséges időért fizet egy bizonyos követelmény helyett.
 - ez hasznot hoz mind az ügyfélnek, mind a fejlesztőnek.
- Az agilis módszertan hol nem alkalmas:
 - nagyobb kaliberű rendszerek fejlesztésére,
 - ahol a fejlesztők különböző helyen dolgoznak,
 - és ahol komplex kölcsönhatások léphetnek fel más hardver- és szoftverrendszerekkel.
 - A módszer nem ajánlott kritikus rendszerek fejlesztésére sem.

23

AZ EXTRÉM PROGRAMOZÁS...

24

Extrém programozás (XP)

- Az XP a legismertebb és legszélesebb körben használt agilis módszer.
- A megközelítés hogyan fejlődött ki?
 - Elismert gyakorlati alkalmazásokat erőltetett
 - Pl.: az iteratív fejlesztés és ügyfél „extrém” részvételének használata.
- **Fő célja:** hogy csökkentse a változások költségvonzatát.

25

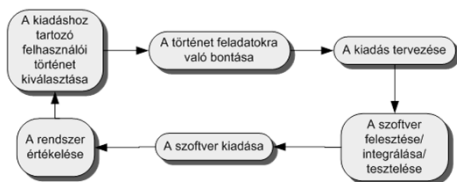
Extrém programozás (XP) jellemzői

- A költségek csökkentését úgy teszi, hogy
 - más alapvető értékeket, elveket, és gyakorlatot vezet be.
- Így az XP-t használó rendszerfejlesztési projekt:
 - sokkal rugalmasabb lesz a röptében bekövetkező változásokkal szemben.
- Minden követelményt **forгатókönyvként** állítanak össze:
 - amely közvetlenül feladatok soraként kerül implementálásra.

26

Extrém programozás (XP)

- **A programozók párokban dolgoznak:**
 - mindenre tesztek készülnek, még mielőtt megírnák a kódot.
 - Minden tesztnek sikeresen le kell futnia, mielőtt az új kódot elhelyeznék a rendszerben.
- **XP kiadási ciklusai:**



27

Extrém programozás is agilis

- Megfelel az agilis módszerek alapelveinek:
 - 1. Az inkrementális fejlesztés a rendszer kisméretű, gyakori kiadásán keresztül valósul meg.
 - 2. Az ügyfél részvételének biztosítása elérhető a fejlesztőcsapatba történő teljes munkaidős bevonásával.
 - Az ügyfél képviselője részt vesz a fejlesztésben és a rendszer elfogadási tesztjeinek meghatározásáért.

28

Extrém programozás is agilis

- 3. A párban való programozás, a rendszerkód fölötti együttes tulajdonjog az embereket támogatja nem folyamatokat.
- 4. A változtatások támogatása
 - a rendszeres rendszerverziók, az előrehozott tesztelést alkalmazó fejlesztés és a folyamatos integráció.
- 5. Az egyszerűség fenntartása
 - a kód minőségének folyamatos tökéletesítéssel történő növelése,
 - valamint az egyszerű tervek használata a rendszer jövőbeli változtatásainak elkerülésére.

29

Extrém programozás (XP)

- Az XP-folyamatban az ügyfél bensőséges szerepet játszik:
 - a rendszerkövetelmények specifikációjában és fontossági sorrendjének meghatározásában.
- A követelményeket nem a szükséges rendszerfunkciók listájaként adják meg,
 - hanem a rendszer megrendelője a fejlesztőcsapat tagjaként megbeszéli a forgatókönyvet a csapat többi tagjával.
- Közösen kidolgoznak egy „történetkártyát”,
 - ami magában foglalja az ügyfél kéréseit.
 - A csapat ezután megpróbálja implementálni a forgatókönyvet a szoftver egy jövőbeli kiadásában.

30

XP - történetkártyák

- Amint a történetkártyák elkészültek:
 - a fejlesztőcsapat feladatokra bontja,
 - megbecsüli az implementáláshoz szükséges időt és erőforrásokat.
- Az ügyfél ezután rangsorolja az implementálandó történeteket,
 - kiválasztva azokat, amelyek azonnal felhasználhatók üzleti célok támogatására.
 - Ha a követelmények változnak: a nem implementált történetek változhatnak vagy elhagyhatók.
- Ha változások szükségesek egy olyan rendszerben, amelyet már átadtak,
 - új történetkártyákat kell készíteni,
 - Ekkor megint csak az ügyfél dönti el, hogy ezen a változásoknak elsőbbségük van-e az új funkcionalitással szemben vagy sem.

31

Extrém programozás (XP) (kiadások)

- Az XP az iteratív fejlesztés „extrém” megközelítését használja.
- Naponta többször megjelenhet a szoftver új verziója,
 - az inkremensek nagyjából **kéthetente** kerülnek az ügyfélnek.
- Amikor létrejön a rendszer egy új verziója:
 - akkor minden létező automatizált teszten végig kell futtatni,
 - beleértve az új funkcionalitásra készült teszteket is.
 - Az új verzió csak akkor elfogadható, ha az összes teszt sikeresen lefutott.

32

Extrém programozás tulajdonságai

- Az XP figyelmen kívül hagyja a szoftvertervezés azon alapelvét,
 - hogy a változtatásokra kell tervezni.
- Az XP szerint ez időpocséklás,
 - mert gyakran nem a várt változtatások, hanem teljesen más követelmények valósulnak majd meg.
- Az előre nem várt változtatások implementálásának problémája:
 - ronthatják a szoftver struktúráját, ami miatt a megvalósíthatóságuk egyre nehezebbé válik.

33

Extrém programozás tulajdonságai

- Az XP úgy oldja meg ezt a problémát,
 - hogy a **szoftver folyamatos kódátszervezését** javasolja.
- **Jelentése:** a programozó csapat tökéletesítési lehetőségeket keres a szoftverben,
 - és azokat azonnal implementálja.
- **Következmény:**
 - a szoftver mindig érthető és könnyen változtatható marad az új történetek implementálását követően is.

34

Köszönöm a figyelmet!

35