Agilis fejlesztés

A vállalatok ma globális, gyorsan változó környezetben működnek. Reagálnak az új lehetőségekre és piacokra, a gazdasági környezet változásaira. A szoftver része minden műveletnek, így 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 a szoftver gyors üzembe helyezése érdekében.

Mivel ezek a vállalatok folyamatosan változó környezetben tevékenykednek, sokszor gyakorlatilag lehetetlen, hogy a szoftverrel szemben stabil elvárásokat fogalmazzunk meg. Ez idézte elő az olyan fejlesztési folyamatokat, amelyek a gyors szoftverfejlesztésre és átadásra összpontosítanak.

A gyors szoftverfejlesztési folyamatokat arra tervezték, hogy segítségükkel gyorsan készíthessünk használható szoftvereket. Ez általában olyan iteratív folyamat, ahol a specifikáció, a tervezés, a fejlesztés és a tesztelés átfedi egymást. A gyors szoftverfejlesztés alapvető jellemzői:

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

Agilis kiáltvány

A klaszikus módszertanokkal való elégedetlenség oda vezetett, hogy a szoftverfejlesztők egy része az 1990-es években új, agilis (jelentése gyors) módszereket indítványozott. Az indítvány célja az volt, hogy a fejlesztőcsapat magára a szoftverre koncentráljon, ne pedig annak tervezésére és dokumentálására.

Az Agilis Fejlesztés egy módszertan-család, és nem egy konkrét megközelítése a szoftverfejlesztésnek.

2001-ben 17 ember, az akkor „pehelysúlyú módszertanok” néven futó módszertanok jelentős képviselői, összeültek, hogy megbeszéljék, mi a közös a módszertanaikban. Megalkották az **Agilis Kiáltvány**t, amit széles körben elfogadtak, mint az agilis fejlesztés kanonikus meghatározását. A szoftverfejlesztés jobb módjait fedezzük fel azáltal, hogy csináljuk, és segítünk másoknak is csinálni. Ennek során az alábbi hangsúly-eltolódásokat találjuk:

- Egyének és interakcióik, szemben az eljárásokkal és eszközökkel.
- Működő szoftver, szemben a teljeskörű dokumentációval.
- Együttműködés az ügyféllel, szemben a szerződésről való alkudozással.
- Változásokra való reagálás, szemben a terv követésével.

Ez azt jelenti, hogy a jobb oldalon szereplő értékek is fontosak, de a bal oldalon lévőket fontosabbnak tartjuk.

Agilis fejlesztés működése

Az agilis szoftverkészítés (agile sw developement) egy elméleti keretrendszer. Többféle agilis fejlesztési eljárás van, de jórészt mindegyik a kis kockázatú fejlesztésre törekszik a rövid idejű fejlesztési ciklusok (ismétlések, iterációk) használatával. Minden ismétlés egy teljes szoftverfejlesztési ciklus: tervezés, követelményelemzés, kivitelezés, kódolás, tesztelés és dokumentálás. Egy ismétlés célja, hogy a ciklus végére egy letesztelt, elérhető kiadás jöjjön létre az alkalmazásból. Minden iteráció végén a fejlesztői csapat újraértékeli a projekt főbb céljait. Az agilis módszerek a szemtől-szembeni kapcsolatot részesítik előnyben az írásos dokumentációk helyett. Emiatt az agilis csoportok jellemzően egyetlen nagy irodában helyezkednek el (pl. scrum). Az agilis módszereknél a haladás legfőbb mérőszáma a működő szoftver.

Az agilis módszerek alapelvei

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

7. A projekteket motivált, megbízható munkatársak vezetik

8. Folyamatos figyelem kíséri a műszaki színvonalat és a tervet

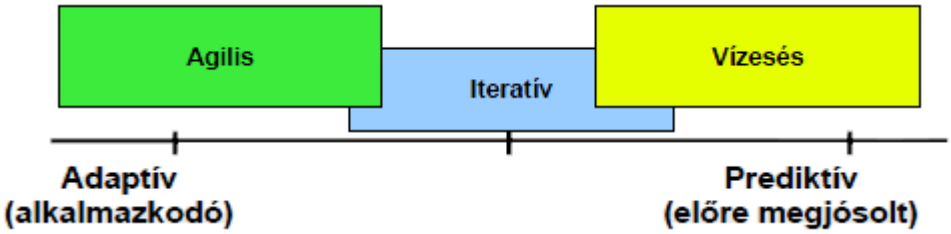
9. Egyszerűség

10. Önszerveződő csapatmunka

11. A változó körülményekhez való gyors alkalmazkodás

Összehasonlítás más módszertanokkal

Az Agilis módszertanokat a „terv-vezéret” vagy „fegyelmezett” módszertanok ellentétének szokták nevezni. Ez félrevezető, mert úgy hangzik, mintha az Agilis módszertanok „tervezetlenek” vagy „fegyelmezetlenek” lennének. Szerencsésebb megkülönböztetés, ha a számegyenes két végpontjának az „adaptív” és a „kiszámítható” módszertanokat tekintjük. Ebben az esetben az Agilis módszertanok a számegyenes „adaptív” végén lesznek.



Az adaptív módszertanok arra fókuszálnak, hogy a gyakran változó követelményekhez tudjanak alkalmazkodni. Ha egy projektben megváltoznak az igények, akkor egy adaptív csapat képes alkalmazkodni a változásokhoz, viszont nehezen tudja megmondani, hogy mi fog történni a jövőben. Minél távolabbi pontról van szó a jövőben, annál bizonytalanabb lesz az adaptív módszertan elképzelése arról, hogy mi is fog akkor történni.

A kiszámítható módszertanok ennek ellenkezőjeként, arra fókuszálnak, hogy minél részletesebben megtervezzék a jövőt. Egy kiszámítható csapat pontosan meg tudja mondani bármelyik pillanatban, hogy mikor milyen feladatok és feature-ök lesznek készen a projektben. A prediktív csapatok nehezen váltanak irányt. A terv általában az eredetileg meghatározott cél elérésére van optimizálva, és az irányváltoztatás könnyen azzal a következménnyel járhat, hogy az elkészült munkát el kell dobni, és újrakezdeni.

Kritikák

Rengeteg kritika olvasható ezekről a módszertanokról, aminek részben az az oka, hogy ezek többségükben még nem kiforrott, lezárt módszertanok, ráadásul mindenki a maga szája íze szerint értelmezi őket. A jogosnak elismert kritikák három központi érv köré gyűlnek:

1. Csak gyakorlott, szenior fejlesztőkkel működik.

2. Nincs eléggé megtervezve a szoftver.

3. Túl sok kulturális változás kell ahhoz, hogy jól működjön.

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:infrendalapjai_architekturak:szoftvertechnologia:agilis_fejlesztes

Last update: **2024/11/13 19:39**