



Agilný manifest: Jednotlivec a jeho tím plánujú spoločne

Agilné metódy sa začiatkom 90. rokov objavili vo vývoji softvéru a okolo roku 1999 získali prvú popularitu s extrémnym programmingom (Kent Beck). Význam agilných metód v posledných rokoch výrazne narastá. To platí predovšetkým – ale nie výlučne – pre IT sektor: 62 % projektov vývoja softvéru pôsobí v Nemecku čiastočne agilne, 16 % dokonca výhradne, zatiaľ čo len 22 % všetkých projektov pracuje výlučne s konvenčnými metódami.

Agilná filozofia zdôrazňuje schopnosť jednotlivca a samoorganizovanie vývojového tímu. Prostredníctvom „agilných metód“ sa usiluje o odstránenie nedostatkov predchádzajúcich modelov vývoja softvéru, ktoré sú vnímané ako príliš strnulé. Keďže strulé a priebežné plánovanie už nie je praktické, nie sú požiadavky na softvér pevne stanovené už od začiatku. Namiesto toho sa vyvíjajú s rastúcim porozumením pre softvérový systém, čo sa má dosiahnuť hlavne intenzívnou komunikáciou so zákazníkom. Na uľahčenie komunikácie sa okrem iného využíva extrémny programming pre porozumenie tzv. „user stories“, čiže príbehov užívateľov na opísanie systémových vlastností, resp. podstatných funkcií. Plánovanie prebieha v kooperácii a nie je samostatným rozhodnutím projektového manažéra. Súhra skupiny pri plánovaní rozsahu ďalšieho schválenia (softvéru) alebo poskytnutia sa označuje ako „planning game“. Planning games a odhad rozsahu projektu sa cyklicky opakujú (iteratívny proces).

Na stretnutí v Utahu v roku 2001 bol pre tento druh vývoja softvéru zvolený pojem „agilný“ a formulovaný „agilný manifest“ (www.agilemanifesto.org/iso/de/manifesto.html):

„Otvárame lepšie cesty vývoja softvéru tak, že to robíme sami a pritom **pomáhame** druhým.

Pri tejto činnosti sme sa naučili ceniť si tieto hodnoty:

- **Indivídua a interakcie** sú nadradené procesom a nástrojom.
- **Fungujúci softvér** je nadradený nad obsiahlou dokumentáciou.
- **Spolupráca so zákazníkom** má prednosť pred vyjednaním zmluvy.
- **Reagovanie na zmeny** má prednosť pred dodržiavaním plánu.“

Agilné princípy sú hlavnými zásadami agilnej práce. Niekedy sa agilné princípy označujú aj ako metóda. **V agilnom manifeste je uvedených 12 princíпов.**¹

- Uspokojenie zákazníka včasnou a kontinuálnou dodávkou hodnotného softvéru
- Agilné procesy využívajú zmeny (dokonca i neskoršej fáze vývoja) ako konkurenčnú výhodu zákazníka.
- Dodávanie fungujúceho softvéru v pravidelných, prednostne krátkych intervalov (niekoľko týždňov alebo mesiacov)
- Takmer každodenná spolupráca odborníkov a vývojárov počas projektu (napr.: spoločný prístupový kód (Collective Code Ownership))
- Poskytovanie prostredia a podpory, ktorú potrebujú motivovaní jednotlivci na plnenie úloh
- Prenos informácií podľa možnosti v osobnom rozhovore
- Za najdôležitejšiu mieru pokroku sa považuje funkčnosť softvéru.
- Dodržiavanie rovnomerného pracovného tempa na strane zadávateľov, vývojárov a užívateľov v záujme trvalo udržateľného vývoja
- Stále venovanie pozornosti technickej excelentnosti a dobrého dizajnu
- Jednoduchosť je esenciálna.

- Samoorganizácia tímu pri plánovaní a realizácii
- Samoreflexia tímu vzhľadom na vlastné správanie pri prispôsobení s cieľom zvyšovania efektivity

Princíp trvalo udržateľného tempa: žiadne nadčasy

) MAINTAIN A SUSTAINABLE PACE



Jedným z 12 agilných princípov je dodržiavanie trvalo udržateľného tempa (sustainable pace). U Becka je „40-hour week“ (40-hodinový týždeň) jednou z 12 praktík extrémneho programmingu. Nadčasy sú signálom pre závažné problémy projektu, ktoré sa musia lokalizovať a vyriešiť.

Pretože v dôsledku neustálych nadčasov vývojári takmer nie sú schopní generovať jasné a prehľadné kódy. Sú neproduktívni a programujú viac chýb. Roman Pichler ako prominentný zástupca to opisuje nasledovne: V „Scrum“ – najrozšírenejšej agilnej metóde – sú pravidelné nadčasy neprijateľné.

»Agilný vývoj softvéru je súhrnným pojmom pre celý rad metód a praktík založených na hodnotách a princípoch manifestu agilného vývoja softvéru.« (Agile Alliance, 2018)

V metodike **Scrum**, najrozšírenejšom agilnom pracovnom postupe, existujú roly Product owner, Team a Scrum master. Product owner zaujíma perspektívu zákazníka a opisuje požiadavky, ktoré musia byť splnené, aby sa dosiahol cieľ projektu, v tzv. product backlogu. Product owner spracováva tento dokument kontinuálne a prioritizuje jednotlivé požiadavky v závislosti od celkového pokroku.

Požiadavky sa realizujú v tzv. stále rovnakých sprints (2 až 4 týždne). V sprint planningu sa vývojový tím dohodne s product ownerom na platných požiadavkách. Tím volí úlohy, ktorým sa bude venovať v ďalšom sprinte. Odhaduje s tým spojenú náročnosť, aby sa zaručilo, že sa úlohy budú dať zvládnuť v stanovenom intervale sprintu.

Výsledky sprintu sa potom v sprint review predstavia product ownerovi (príp. aj zákazníkom/užívateľom). Za uzavreté sa považujú výlučne otestované a ‚prevádzky schopné‘ funkcionality.

Tím sa počas celého procesu organizuje úplne samostatne a musí mať k dispozícii príslušné zdroje a schopnosti. Retrospektívne tím po sprinte reflektuje pracovný proces a spoluprácu, aby určil zlepšovacie opatrenia. Scrum master ako expert pre danú metódu podporuje tím pri dodržiavaní pravidiel, ktoré si sám stanovil, a odstraňovaní prípadných prekážok.

Dr. Nadine Müller, ver.di (DE)