

PETER HLADKÝ, RIADITEĽ VÝVOJA SOFTVÉRU POSAM

## AGILE PRAGMATICKY (ČASŤ 1) ZMENA PARADIGMY

Softvér je v mnohých odvetviach základným stavebným prvkom hodnototvorných procesov firmy. S rozširovaním softvérových systémov a ich pokrytia biznis procesov rastie aj komplexita systémov samotných, ako aj prostredia do ktorého nový systém integrujeme. Frekvencia zmien sa zrýchľuje, miera neurčitosti a vzájomného ovplyvňovania rôznymi faktormi rastie. Je nesporné, že práca softvérových špecialistov má rastúci význam a úspešnosť dodávateľa pri implementácii softvérového diela priamo ovplyvňuje hospodársky výsledok zákazníka.

### ZMENA PRÍSTUPU PRI VÝVOJI SOFTVÉRU

Implementovať systém do zložitého prostredia je ťažké aj za predpokladu veľmi presnej predstavy zákazníka o cieľovej podobe systému. Realitou sú však často neúplné, alebo vnútorne neskonsolidované, požiadavky zákazníka. Ďalším fenoménom je čas. Biznis tlačí na rýchlosť dodávky a IT, ak má biznis podporovať, musí hľadať cesty ako túto požiadavku zabezpečiť.

V takýchto podmienkach klasický systems engineering proces: Zber požiadaviek – Analýza – Design – Implementácia – Verifikácia, známy tiež ako vodopádový, prestáva fungovať. Treba hľadať nové prístupy, reflektujúce zmenenú realitu. V PosAm sme adoptovali agilné metodiky vývoja. Začali sme so Scrumom a postupne sme sa v niektorých projektoch dostali ku Kanbanu.

### PREČO MÁ AGILE VÄČŠIE PREDPOKLADY RIEŠIŤ KOMPLEXNÉ SYSTÉMY?

Riadenie dodávky zákaznickeho riešenia je vo svojej podstate riadenie komplexného adaptívneho systému. Navzájom sa v ňom ovplyvňuje niekoľko oblastí: zákazníci, používatelia, vývojový team, rozpočet, technológia, scope. Každá z týchto oblastí je komplexnou problematikou sama o sebe, nielen ešte vo vzájomnej interakcii.

V takomto systéme je výstup menej predstaviteľný ako výsledok sady príčinnonásledných udalostí, lebo takto jednoducho lineárne realita nefunguje. Skôr si riadenie vývoja takéhoto systému treba predstavovať pomocou abstrakcie z oblastí technickej

kybernetiky a systémov riadenia, ako navzájom previazané spätnoväzobné slučky, ktoré jedna na druhú pôsobia.

Agile je pre tento typ systémov vhodnou voľbou z nasledujúcich dôvodov:

Agile je feedback-driven, ďaleko viac ako plan-driven. Stavia na fyzickom prístupe k doménovému expertovi. Namiesto spracovávaní rozsiahlej dokumentácie stavia na častom poskytovaní spätnej väzby a na princípe ALAP (tak neskoro ako sa len dá) v prijímaní rozhodnutí. V PosAm sa nám osvedčil aj test-driven prístup, kedy si spätnú väzbu nechávame poskytovať regresne od všetkých testov pripravených doteraz ku existujúcim modulom a aplikujeme potrebné zmeny tak, aby testy stále fungovali.

Agile počíta so zmenou a nebráni sa jej - čo je v kontraste s vodopádovou metodikou. Dôležité je rozhodnutie zákazníka (alebo analytika v roli produktového vlastníka ak ho na projekte zastupuje), že zmena lepšie reprezentuje ciele, ktoré chce zákazník riešením dosiahnuť. Dopady zmeny musíme identifikovať a dohodnúť sa na mitigácii jej dopadov.

Agile funguje inkrementálne, na základe vyžadovanej a dodávanej spätnej väzby. V PosAm nám dobre funguje prístup formou tzv. „walking skeleton“ – kedy si na začiatku postavíme kostru prostredia, zatiaľ chýbajúce moduly nahradíme ich umelými napodobeninami (tzv. „mock“-mi) a overujeme fungovanie ranných verzií softvéru v simulovanom prostredí, ktoré postupne nahradzujeme reálnymi komponentami.

## OBMEDZENIA PRI APLIKOVANÍ AGILE

Podoba zmlúv medzi dodávateľom a odberateľom je veľmi dôležitým limitujúcim faktorom pri aplikovaní agile. V našich zemepisných šírkach je stále prevládajúcim typom kontraktu „fix price“. Ak uzatvoríme zmluvu na dodávku diela s definovaným setom požiadaviek, termínom a cenou, je šanca fungovať so zákazníkom agilne obmedzená. Nie je však nemožná.

Ak aj zo zmluvy vyplýva, že scope je daný, vieme s ním pracovať formou zmenového riadenia. A tu nastupuje rozmer dôvery a transparentnosti, ktoré sa vytvárajú dlhodobo a je veľmi náročné ich dosiahnuť u nového zákazníka v neznámom prostredí.

**Vplyv neurčitosti** – či už funkčnej alebo technologickej, je ďalším obmedzením pre aplikovanie agile. Proste v okamihu podpisu zmluvy nie je úplne jasné AKO a/alebo ČO sa má vlastne dodať. A možno niekedy ani nie je jasné čo nie je jasné.

Tu aj pre agile fungovanie platia zákonitosti riadenia rizika a musíme zaviesť mitigačné opatrenia na elimináciu rizík. Jedným kľúčových opatrení je prototypovanie najdôležitejších doménových a technologických konceptov tak skoro ako sa len dá.

**Vplyv veľkosti** – ideálna veľkosť agilného tímu je 7+/-2. Veľké enterprise projekty bez problémov túto hranicu prekročia. Agilné fungovanie v takomto prípade sa nezaobíde bez paralelizácie na viacero tímov a zvýšeného dôrazu na „ceremónie“ v oblasti plánovania, vzájomnej komunikácie a integrácie.

Dostatočná analýza a design, spolu s high-level architektúrou, sú v tomto prípade dôležitým komunikačným prostriedkom medzi teamami a ich príprave je potrebné venovať potrebný čas a prostriedky.

## SKÔR AKO SA PUSTÍM DO DRUHEJ ČASTI

Agile prístup k vývoju softvéru, má svojim inkrementálnym, feedback-driven prístupom, ktorý počíta so zmenou, schopnosť reflektovať realitu pri vývoji a nasadzovaní nových IT systémov. Prax však prináša obmedzenia, s ktorými sa treba aj pri agile vývoji v praxi vysporiadať. Ako to riešime v PosAm, o tom bude nasledujúca časť.

### Čo nájdete v ďalších častiach:

Agile pragmaticky (časť 2)  
Best practices, alebo aplikácia agile v praxi

Agile pragmaticky (časť 3)  
Softvér nepíšu stroje, ale ľudia