PosAm USEFUL TECHNOLOGIES

PETER HLADKÝ, RIADITEĽ VÝVOJA SOFTVÉRU POSAM

AGILE PRAGMATICKY (ČASŤ 2) BEST PRACTICIES, ALEBO APLIKÁCIA AGILE V PRAXI

V prvej časti Agile pragmaticky - Zmena paradigmy, som sa venoval príčinám, ktoré vedú IT spoločnosti k zmene prístupu pri vývoji softvérových diel. Zároveň som poukázal na spôsob, ako zabezpečiť vyššiu efektivitu delivery postupov. Je ním zmena metodík pri vývoji softvéru. V PosAm sme adoptovali agile. Jeho výhody pri reflektovaní dnešnej reality pri vývoji systémov sme pomenovali a ukázali na ich prínosy. Všetko má ale svoje Ale.

AKO SA VYSPORIADAŤ S OBMEDZENIAMI PRI APLIKOVANÍ AGILE

Ako som spomenul v závere prvej časti, na aplikovanie agile negatívne vplývajú tri faktory. Prvým je podoba zmlúv medzi dodávateľom a odberateľom, druhým faktorom je vplyv neurčitosti a tretím je veľkosť projektu.

Cestou ako sa vysporiadať so zmluvne pevne definovaným scope môže byť prioritizácia backlogu požiadaviek a systém ich zámeny. Pod prioritizáciu chápeme usporiadanie požiadaviek na riešenie takým spôsobom, že prvé sú uvedené tie najdôležitejšie pre biznis zákazníka alebo najproblematickejšie či už z vecného alebo technologického hľadiska.

Pre eliminovanie rizík spôsobených vplyvom neurčitosti zadania je dôležité prototypovanie najdôležitejších doménových a technologických konceptov tak skoro ako sa len dá. Prioritizácia backlogu však rieši tento problém iba čiastočne. Nestotožňujem sa s názorom, že backlog je možné pripraviť rýchlo pre komplexný systém. Čím je riešená problematika komplexnejšia tým dlhšiu prípravu backlogu si vyžaduje. Môže predstavovať celú fázu v životnom cykle projektu.

DOBRÉ UCHOPENIE PROBLEMATIKY VĎAKA HLD

V internej metodike PosAm sme zaviedli koncept HLD (High Level Design) ako pomenovanie prípravnej fázy v dodávke SW riešenia. V tejto fáze sa snažíme identifikovať explicitné aj implicitné požiadavky do backlogu a navrhnúť konceptuálnu architektúru riešenia. Sem patrí aj prototypovanie, UX design a analýza smerom do šírky s cieľom pochopiť šírku domény (chceme pochopiť základnú dekompozíciu riešenej problematiky bez potreby riešiť všetky detaily).

Naša skúsenosť potvrdzuje platnosť tzv. Conweyovho zákona, že sofvérový dizajn je väčšinou poplatný komunikačnej štruktúre projektu alebo organizácie. Ako scope projektu rozdelíme do funkčných celkov na začiatku a rozdelíme zodpovednosť teamom, tak akceptujúc toto rozdelenie má tendenciu vzniknúť aj architektúra a štruktúra modulov systému. Preto je dôležité, aby toto delenie vzniklo tak neskoro ako sa len dá a predchádzala mu prípravná fáza HLD (inak sa môže stať, že rozdelenie nebude rešpektovať siločiary riešenej problematiky). Za túto fázu preto u nás nezodpovedá projektový manager, ale solution architekt.

DOMAIN DRIVEN DESIGN POMÁHA, KEĎ JEDINOU ISTOTOU JE ZMENA

Zmeny určite nastanú, to je jediná istota na ktorú sa určite vieme spoľahnúť. Naše skúsenosti hovoria, že zmeny sa ľahko robia do systému, ktorý pri dekompozícii kládol dôraz na vysokú kohéznosť modulárnej dekompozície, t.j. také rozdelenie, kedy je jedna oblasť problematiky ucelene obsiahnutá v príslušnom module a ako následok dochádza ku minimalizácii vzájomných závislostí medzi modulmi navzájom. Toto nemožno

dosiahnuť bez dostatočného spoznania domény či už v technologickej, ako aj funkčnej oblasti. Preto sme do toolkitu znalostí v PosAm zaradili techniku Domain Driven Design (DDD).

DDD je metodika, ktorá využíva na prípravu designu riešenia jednotný jazyk medzi biznisom a vývojármi s cieľom identifikovať core doménu a súvisiace subdomény ako samostatné moduly. Používa techniky taktickej (úroveň tried) a strategickej (úroveň modulov) dekompozície, aby čo najlepšie premietla vecnú problematiku do softvérového kódu.

DÔLEŽITOSŤ SPÄTNEJ VÄZBY PRE ZACHOVANIE INTEGRITY RIEŠENIA

Dekompozícia a modularita so sebou prinášajú jeden problém. Každá dekompozícia je forma zjednodušenia, aby som veľký problém vedel uchopiť. Ak modul/suboblasť vnímam a do značnej miery zaobchádzam s ňou ako s nezávislou časťou projektu, môže dôjsť ku skresleniu, a to tak, že táto časť síce sama o sebe funguje podľa zadania, ale ako súčasť celku nám ovplyvňuje výsledné správanie systému negatívnym spôsobom. Preto je dôležité neustále zavádzať do procesov prvky spätnej väzby a komunikovať medzi tímami.

Dobrým začiatkom budovania systému spätnej väzby smerom od infraštruktúry a technológie je postavenie tzv. „walking skeletonu“. Jedná sa o kostru riešenia, ktorá využíva pokiaľ možno všetky technologické komponenty cieľového riešenia, vrátane automatizácie zostavovania a nasadzovania . Cieľom je pripraviť si pôdu pre časté nasadzovanie releasov a tým častú spätnú väzbu o kvalite pripraveného kódu. Koncept DoD (definition of done) definujúci kedy je požiadavka hotová nám spolu s inkrementálnym rozširovaním kostry riešenia umožňuje neustálu spätnú väzbu.

SKÔR AKO SA PUSTÍME DO TRETEJ ČASTI

Prioritizácia backlogu, High Level Design, Domain Driven Design, Budovanie „walking skeletonu“, Koncept Definition of Done. To sú základné prístupy ktoré v PosAm využívame pre zabezpečenie úspešnosti komplexných projektov s nie celkom jasne definovaným zadaním. Ale aby to všetko fungovalo, je tu ešte jeden veľmi dôležitý faktor, a tým sú ľudia. Ale o tom až v ďalšej časti.

Čo nájdete v ďalších častiach:

Agile pragmaticky (časť 1)
Zmena paradigmy

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

PosAm, spol. s r. o.
Bajkalská 28
821 09 Bratislava
T: +421-2-49 23 91 11
www.posam.sk