Systematické výmeny zamestnancov

MAREK FUČILA

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava

Abstrakt. Ak sa v konkurenčnom prostredí prestáva spoločnosti dariť a chce sa udržať na trhu, musí znižovať náklady. Spravidla dochádza k veľkému prepúšťaniu zamestnancov. Takéto riešenie je však nepopulárne a navyše skúsenosti ukazujú, že zamestnanci sú aj po ozdravnom procese schopní identifikovať slabších spolupracovníkov, ktorí ich brzdia.
Táto esej sa venuje systematickému prístupu k výmene zamestnancov. Takýto prístup uplatňujú s rôznymi obmenami napríklad v spoločnostiach General Electric, Toyota či Microsoft. Ide vlastne o vyraďovanie najslabších zamestnancov na základe hodnotenia ich výkonov. Vychádza sa z výskumov, ktoré ukazujú, že sú až priepastné rozdiely v efektivite práce jednotlivcov, a preto je vhodné sa zbavovať najslabších pracovníkov postupne. Najlepších zamestnancov je možné vhodne motivovať, a tak sa každý rok zvyšuje efektivita práce danej organizácie.
Problémom stratégie je hlavne otázka korektnej klasifikácie zamestnancov. Väčšie softvérové spoločnosti totiž tvoria tímy vedené rôznymi manažérmi s rôznymi kritériami hodnotenia, pričom nie je ľahké dosiahnuť konzistenciu. Niektorí schopní vývojári môžu mať nižšiu produktivitu práce napríklad kvôli zlej organizácii.

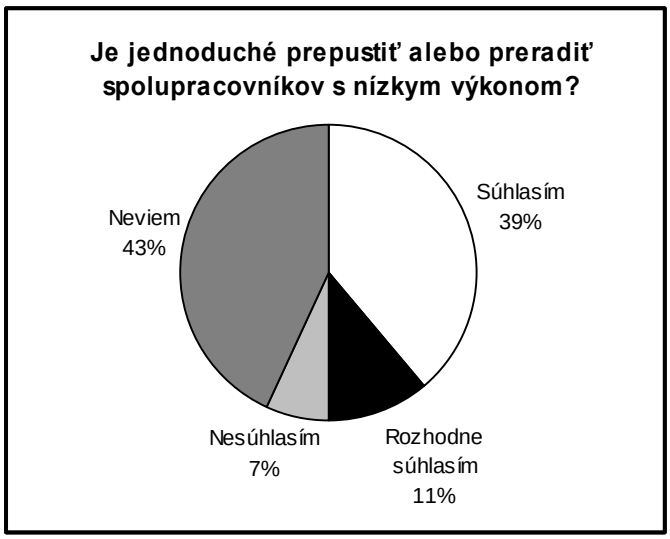
Úvod

Každá spoločnosť stojí a padá na svojich zamestnancoch. Úspešnosť riešenia projektov závisí od efektivity práce tímov ako aj jednotlivých zamestnancov. Ak je efektivita práce príliš nízka, spoločnosť prestáva konkurovať ostatným spoločnostiam na trhu, pretože vyvíja softvér s vyššími nákladmi.

Podstatnú časť nákladov tvoria mzdy zamestnancov. Aby bolo možné znížiť náklady na vývoj a obstáť v konkurenčnom boji, musí spoločnosť v kríze prepúšťať ľudí. Spravidla dochádza k veľkému prepúšťaniu, ktoré nebolo dlhodobo plánované, a tak manažéri obvykle nemajú k dispozícii dostatok informácií, aby vedeli správne vybrať najschopnejších zamestnancov, ktorí ostanú. Potvrdzujú to prieskumy robené formami ankiet a stretnutí s vývojármi, ktorí by mali po masívnom prepúšťaní byť práve tí najlepší.

Manažment v softvérovom inžinierstve, máj 2005, s. 1-6.

Ako Middleton a kolektív uvádzajú vo svojom článku [1], zamestnanci sú aj po takomto ozdravnom procese schopní identifikovať svojich slabších spolupracovníkov. Po dlhšej dobe opäť naberajú pocit, že medzi nimi zbytočne zotrvávajú slabší kolegovia. Autori uvádzajú odpoveď zamestnancov na anketovú otázku *„Je jednoduché prepustiť alebo preradiť spolupracovníkov s nízkym výkonom?“* v grafe uvedenom na Obr. 1.



Obr. 1. Iba polovica vývojárov si myslí, že je jednoduché zbaviť sa pracovíkov s nízkym výkonom

Celoplošné prepúšťanie teda nezabráni opakovanému trvalému poklesu efektivity práce celku, a preto je len otázkou času, kedy nastane potreba ďalšej nepopulárnej reorganizácie spoločnosti. Manažéri, ktorí sa snažia predísť tejto perspektíve využívajú niektorú z foriem systematického prístupu k výmene zamestnancov.

Výskumy

Opodstatnenosť snahy o neustále zvyšovanie efektivity práce inžinierov pri vývoji softvéru dokazujú výskumy. Podľa autorov článku [1], Barry Boehm a Richard Turner zistili, že dobrí ľudia a zohrané tímy tromfnú ostatné faktory. Je dôležité mať kvalitných kvalifikovaných inžinierov, ktorí dokážu spolupracovať so svojimi kolegami. Rovnako je nevyhnutné ich prácu efektívne organizovať a tým celý proces vývoja softvéru zefektívniť. Ak sa nedarí spoluprácu koordinovať, alebo ak sa v tímoch vyskytujú ľudia, ktorí výrazne zaostávajú za kolektívom, a tak brzdia ostatných, degraduje to výkon celku. Barry Boehm a Richard Turner zaznamenali až priepastné rozdiely výkonov jednotlivcov. V rámci jednej softvérovej spoločnosti

môže pomer výkonov rôznych vývojárov podľa ich zistení dosahovať až 26:1, prípadne 10:1 v závislosti od metriky.

Tieto výsledky potvrdzujú aj Tom DeMarco a Tim Lister, ktorí podľa Middletonovho článku [1] taktiež uverejnili údaje ukazujúce rozptyl schopností skúsených softvérových profesionálov približne 10:1. Navyše, zistili aj rozdiel 10:1 v produktivitách rôznych softvérových firiem. To poukazuje na opodstatnenosť úsilia o zlepšovanie procesov vývoja softvéru, s čím súvisí dobrá organizácia práce a premyslené štruktúrovanie tímov. Autori štúdie zistili, že vysoko výkonní zamestnanci robili v organizáciách s presnejším vymedzením úloh, s väčším súkromím a pokojom, pričom trpeli menším počtom prerušení. Každé prerušenie práce totiž znižuje výkon.

Ak sú závery, ktoré nám výskum ponúka správne, potom je logické podporiť systematické výmeny slabých pracovníkov spolu s reorganizáciou tímov v prípade zníženia efektivity práce.

Možné prístupy zvyšovania produktivity

Prístup systematického zvyšovania produktivity práce zamestnancov uplatňujú s rôznymi obmenami napríklad v spoločnostiach General Electric, Microsoft či Toyota.

Jednou z možností, ako udržať vo firme rast produktivity práce, je prístup podľa Jacka Welcha, ktorý zaviedol v General Electric. Tajomstvom úspechu je hodnotenie zamestnancov. Manažéri, ktorí vedú vývojový tím, majú záujem na tom, aby bol celý ich tím dobre hodnotený, pretože je to mierou ich vlastných schopností. V praxi sa preto často stáva, že nadhodnocujú kvalitu práce svojich podriadených, a zároveň zahladzujú nedostatky jednotlivcov. Navonok sa potom zdá, že tím pracuje efektívne, a pritom existujú skryté rezervy. Časom sa problém zhoršuje. Tomu Welch zabránil tak, že uložil manažérom povinnosť hodnotiť svojich podriadených podľa gaussovej krivky. Hodnotenie tak zodpovedá normálnemu rozloženiu schopností zamestnancov. Identifikujú sa tak najslabší členovia tímu, ktorí brzdia prácu ostatných a tým aj celej spoločnosti. Welch stanovil hranicu na najslabších 10 percent. Títo ľudia dostanú šancu zlepšiť sa. K tomu im môžu pomôcť napríklad rôzne školenia. Zamestnanci, ktorí nedokážu obhájiť svoje zlepšenie, musia firmu opustiť. Naopak najlepší pracovníci môžu byť za svoje výkony odmenení. Takýmto spôsobom sa potom darí manažérom posunúť gaussovu krivku hodnotenia smerom k vyššej efektivite a produktivite práce.

V spoločnosti Microsoft sú vývojári softvéru podobne ako v General Electric nútení zlepšovať svoje výkony, pretože pravidelne musí najslabších 5 percent zamestnancov odísť. Zaujímavé je, že takéto množstvo vývojárov obvykle odchádza dobrovoľne. Relatívne malé personálne výmeny pravdepodobne súvisia s kvalitným prijímacím procesom. Microsoft sa stará o motiváciu svojich najlepších ľudí, aby nestrácal aj ich. Na rozdiel od General Electric sa teda Microsoft nemusí veľmi snažiť podporovať zlepšovanie výkonu najslabších pracovníkov. Vývojári softvéru obvykle často striedajú zamestnanie, a kvôli tejto fluktuácii sa treba zamerať najmä na to, aby neodchádzali tí najlepší. Kľúčová je teda motivácia. Napríklad nadštandardné ohodnotenie.

Spoločnosť Toyota Motor Manufacturing v USA má na rozdiel od General Electric či Microsoftu stupňovitý proces zvyšovania pracovných výkonov, a to bez pevných kvót. Zjednodušene sa dá hlavná myšlienka tejto stratégie opísať sloganom: „ponechať, premiestniť, odstrániť".

Pri zvyšovaní produktivity práce jednotlivcov môže napomôcť systém navrhnutým Wattsom Humphreyom [3] nazvaný Osobný softvérový proces (Personal Software Process - PSP), z ktorého bol odvodený Tímový softvérový proces (Team Software Process - TSP) určený pre použitie vo veľkých projektoch, ktorý je možné využiť na úrovni celého tímu.

Proces PSP je založený na nasledujúcich princípoch:

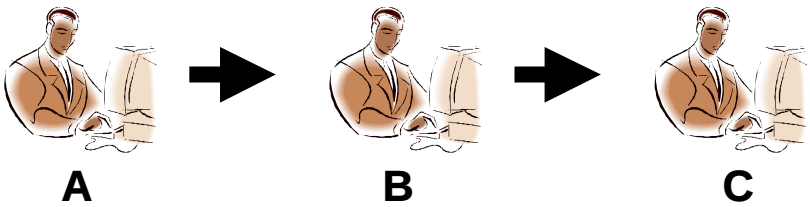
- Každý zamestnanec je iný. Aby mohli pracovať čo najefektívnejšie, musia si stanoviť plány a tie musia byť založené na údajoch, ktoré získali počas práce na predchádzajúcich projektoch.
- Na dôsledné zlepšovanie výkonnosti zamestnancov je nevyhnutné aby používali dobre definované a merateľné procesy.
- Aby sa mohli vyvíjať kvalitné produkty, je potrebné aby zamestnanci cítili osobnú zodpovednosť za kvalitu svojich výrobkov. Zamestnanci sa musia snažiť vytvárať kvalitné produkty.
- Nájdenie a oprava chyby na začiatku procesu je lacnejšia ako v neskorších fázach.
- Je efektívnejšie predchádzať chybám ako ich hľadať a odstraňovať.
- Správny spôsob ako vykonať prácu je taký, ktorý je najrýchlejší a najlacnejší.

Problémy

Využiť proces gaussovej krivky bolo v General Electric každým rokom ťažšie, pretože ostávali len schopní zamestnanci. Tiež nebolo jednoduché použiť ho konzistentne v celej spoločnosti, pretože niektoré odvetvia boli lepšie ako iné. Napriek tomu sa v každoročnom anonymnom zamestnaneckom prieskume zamestnanci tejto spoločnosti stále sťažovali, že očistný proces nebol implementovaný dosť prísne. Tento záver prekvapil samého Welcha, ktorý tvrdo bojoval za to, aby jeho manažéri túto iniciatívu implementovali.

Zamestnanci boli kritickejší k svojim slabším kolegom ako ich manažéri. Možným vysvetlením je výskum dynamiky softvérových projektov Tareka Abdel-Hamida a Stuarta Madnicka, spomínaný v Middletonovej práci [1]. Tento výskum poukazuje pomocou formálneho modelu procesu vývoja softvéru na problémy riešenia ohraničených úloh. S využitím teórie spracovania úloh v rade autori identifikovali slabé miesto, kde dochádza k znižovaniu výkonu tímu.

Príkladom je práca troch vývojárov v tíme. Na obrázku č.2 sú označení ako A, B a C.



Obr. 2. Vývojár B sa môže ocitnúť v pasci

Ak vývojár B pracuje nepretržite, ale dostáva prácu od vývojára A nepravidelne a vývojár C pracuje tiež nevyrovnane, výsledok celého tímu tým bude trpieť. V praxi je nemožné pre vývojára B zlepšiť produktivitu tímu bez toho, aby sa zlepšili ľudia pred a za ním. Autori túto skutočnosť potvrdili simuláciou.

Simulácia ďalej ukázala, že ak je práca dodávaná vývojárovi B príliš variabilne, teda veľké časti práce prichádzajú nepravidelne, produktivita samotného vývojára B klesne. Ak je však také isté množstvo práce dodávané v priebehu rovnakého času s menšími odchýlkami a v menších častiach, potom sa produktivita vývojára B značne zlepší.

Bez zmeny pracovných postupov konkrétneho pracovníka sa tak môže stať, že stúpa alebo klesá jeho produktivita, a to len kvôli vyrovnaným respektíve nerovnomerným dodávkam práce. Ak to nadriadený nepostrehne, mylne tým identifikuje vývojára, ktorého výkon poklesol ako slabý článok reťaze. Zamestnanec v pasci síce môže dostať možnosť zlepšiť sa, to mu však bez zmeny organizácie práce tímu takmer vôbec nepomôže, nakoľko nedokáže ovplyvniť čas nečinnosti, keď nedostal výstupy práce vývojára v rade pred ním.

Záver

Systematické výmeny zamestnancov vyzerajú byť vhodnou prevenciou pred znižovaním produktivity práce spoločnosti. Zvyšovanie výkonu tímov ale nemôže byť postavené len na hodnotení zamestnancov zhora. Samotní vývojári totiž vedia medzi sebou určiť zdroj problémov lepšie ako ich manažéri. Preto by manažéri mali byť schopní na základe podnetov svojich podriadených reorganizovať prácu v tíme, pridelenie úloh, komunikáciu, a tým zvýšiť efektivitu práce jednotlivcov, čím stúpne produktivita tímu. Gaussova krivka hodnotenia totiž odráža reálny stav len pri optimálne zohranom tíme.

Použitá literatúra

1. Middleton, P., Lee, H.W., Irani, S.A.: Why Culling Software Colleagues Is Popular. In IEEE Software, Vol. 21, No. 5 (September/October 2004), 28-32.
2. Madachy, R.J.: Software Process Dynamics - Portions from 4/03 Draft Version, IEEE Computer Society Press (1999)
3. Humphrey, W.S.: The Personal Software Process. CMU/SEI Technical Report 2000-TR-022, November 2000

Annotation

Systematic stuff culling

If the company in the competitive environment goes into a recession and it wants to stay in the marketplace, it has to reduce costs. Commonly a big firing is coming up. But this solution is not popular and the experience shows personnel are able to identify poor colleagues also after this cleaning process.
This essay is concerned about systematic stuff culling. This approach is variedly used in companies like General Electric, Toyota and Microsoft. It is about stuff culling based on ranking their productivity. Studies show us there are huge differences in productivity of stuff members. Therefore it is desirable to cull out weakest workers systematically. It is also worthy to motivate best workers. The productivity of work in the organization is than higher year by year.
The main problem of this strategy is the appropriate classification of stuff. Big software companies composite of teams leaded by different managers with different criteria. It is hard to achieve consistency. Some skilled developers may have lower productivity for bad work organization.