

Zlepšovanie produktivity softvérových tímov

PETER SIVÁK

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
sivak01@student.fiit.stuba.sk*

Abstrakt. Produktivita je definovaná ako pomer veľkosti výstupu v podobe tovarov a služieb a množstva vstupov potrebných na vytvorenie tohto výstupu. Je veľmi dôležité, aby manažéri v softvérových projektoch dobre poznali metódy, ktoré vedú k významnému zlepšeniu produktivity. V eseji opisujem niektoré z týchto metód použitím systematického prístupu k ich kategorizácii.

Úvod

Vývoj softvérových systémov je podobne, ako veľká časť iného ľudského snaženia, výrazne ovplyvnený ekonomickými parametrami. V oblasti vývoja týchto systémov môžu byť úspešné iba spoločnosti, ktoré dokážu minimalizovať čas potrebný na pretransformovanie úvodnej myšlienky do softvérového systému a dodať výsledok v čo najkratšom čase. Osobitne dôležitou je produktivita v jasne definovaných projektoch, v ktorých sa spoločnosti medzi sebou nedokážu výraznejšie odlíšiť iným parametrom. Snahou manažmentov softvérových spoločností sa preto nevyhnutne stáva *kontinuálne zvyšovanie produktivity* svojich tímov.

Produktivita je priamo prepojená so životným štandardom softvérových inžinierov. Ak sa zvyšuje zisk vytvorený za jednu hodinu práce, zvyšujú sa aj príjmy. Ak produktivita klesá, príjmy sa tiež znižujú. V prípade, že sa konkurujúce spoločnosti vo všetkých iných aspektoch vyrovnajú, tak produktívnejšia spoločnosť bude mať vyšší zisk. Kľúčom k pretrvávajúcemu rastu miezd v softvérovom priemysle je preto aj každoročné zvyšovanie produktivity.

Cieľom tejto eseje je zhrnúť viacero metód na systematické zvyšovanie produktivity pri vývoji softvérových systémov. Produktivita sa všeobecne definuje ako *množstvo vytvoreného výstupu za jednotku času*. V komerčnom prostredí spoločností, ktoré vyvíjajú a predávajú softvér ako produkt, sa produktivita často definuje ako vygenerovaný zisk na jednu hodinu práce zamestnanca alebo ako vygenerovaný zisk na každú investovanú korunu. Spoločnosti, ktoré vyvíjajú softvér pre svoju potrebu,

môžu pod produktivitou rozumieť priemerné zvýšenie zisku spoločnosti dosiahnuté z jednej pracovnej hodiny vývojára.

K zlepšovaniu produktivity môžeme pristupovať v zásade dvomi spôsobmi. Jeden je založený na jej priamom meraní. Meranie produktivity nám umožňuje hocikedy objektívne posúdiť, či sa aplikované zmeny prejavili na výslednej produktivite pozitívne alebo negatívne, a následne podľa týchto zistení robiť ďalšie úpravy. Druhý spôsob počíta s používaním existujúcich najlepších postupov v danej oblasti a nepožaduje dôsledné vyhodnocovanie produktivity. Tento spôsob je užitočný najmä v malých softvérových spoločnostiach. Väčšie spoločnosti by mali svoju produktivitu vedieť vyčíslieť.

Meranie produktivity

Ak chceme zvyšovať produktivitu softvérových tímov alebo ich členov je dobré, ak poznáme jej aktuálny stav a trend vývoja. Spoľahlivé meranie produktivity je veľmi náročné a často až nemožné. Dlhú dobu sa v softvérom priemysle používalo meranie produktivity podľa počtu riadkov vo výslednom kóde (LOC; lines of code). Viedlo to samozrejme k umelému naťahovaniu programov o riadky, ktoré sú zbytočné. Táto metóda neumožňuje rozlíšiť neproduktívne činnosti, ako napríklad písanie testovacích zdrojových kódov. Známi je tiež výrok Edsgera Dijkstru, ktorý hovorí, že ak chceme počítať riadky v kóde, nemali by sme ich považovať za riadky vyprodukované, ale za riadky minuté.

Pri meraní produktivity si tiež treba uvedomiť, že súčasťou softvérových tímov nie sú iba programátori. Je potrebné merať aj produktivitu ďalších členov tímu – analytikov, návrhárov a pod. Na úspešné meranie produktivity potrebujeme metriku, ktorá nepodporuje vývojárov v produkovaní množstva kódu bez skutočnej hodnoty. Jednou možnosťou je merať počet implementovaných funkčných celkov, prípadne merať priamo zisk vygenerovaný za hodinu práce vývojára. Použitie posledne menovanej metódy je možné iba v špecifických prípadoch – napríklad pri údržbe systémov.

Myslím si, že dobrým spôsobom je merať produktivitu programátorov podľa počtu vyriešených úloh. Musí však byť splnená podmienka, že navrhnuté úlohy sú dostatočne malé, aby sa každá úloha dala ohodnotiť niekoľkostupňovou stupnicou – napríklad: málo náročná, stredne náročná a veľmi náročná úloha. Ku každému stupňu sa určia váhy. Sčítaním váh všetkých vykonaných úloh za nejaký čas sa získa číselné vyjadrenie produktivity vývojára. Rozbitie veľkých úloh do menších okrem toho pomôže aj pri plánovaní softvérového projektu.

Zlepšovanie produktivity

Predtým, ako preskúmame rôzne metódy na zlepšovanie produktivity samotných softvérových tímov, pozrime sa na to, ako vplýva štruktúra softvérových projektov v spoločnosti na jej celkovú produktivitu, v zmysle veľkosti vytvoreného zisku

v pomere ku vstupným nákladom. Projekty sa v komerčnej sfére spravidla rozdeľujú na také, ktorých cieľom je vytvorenie nového systému, a na také, ktoré udržiavajú existujúce systémy. Podľa [5] je rozumnejšie orientovať spoločnosť viac na vyvíjanie nových systémov, ako na udržiavanie existujúcich. Spravidla vyšší zisk pri vývoji nových produktov je dôsledkom toho, že sa v nich často zhmotňujú nové a inovatívne riešenia, ktorých cena sa stanovuje prvýkrát. Na druhej strane, pôvodné systémy sú postavené na technológiách, ktoré sú náročnejšie na údržbu.

Produktivitu zlepší zníženie množstva vstupov a/alebo zvýšenie množstva výstupov. Množstvo vstupov môžeme zmenšiť znížením miezd vývojárov alebo vykonávaním menšieho množstva práce. Najschodnejšie je vykonávanie menšieho množstva práce tým, že budeme vytvárať iba to, čo zákazník skutočne potrebuje. Zo štatistík [5] totiž vyplýva, že osemdesiat percent hodnoty väčšiny systémov je obsiahnutých v dvadsiatich percentách funkcií a až dve tretiny funkcií systémov je používaných veľmi málo, prípadne vôbec. Aby sme vytvárali naozaj iba to, čo zákazník potrebuje, môžeme ho napríklad motivovať v tom, aby formuloval čo najviac požiadaviek na nový systém už v prvých fázach projektu.

Prejdime teraz k metódam na zvyšovanie produktivity softvérových tímov. Podľa R. Chianga a V. Mookerjeeho [2] si zvyšovanie produktivity vývoja softvéru vyžaduje vyvážený prístup k trom základným pilierom manažmentu v softvérovom inžinierstve:

- k technológiám,
- k ľuďom a
- k procesu vývoja.

V deväťdesiatych rokoch sa produktivita zvyšovala najmä zlepšovaním technológií používaných pri vývoji softvérových systémov. Rovnako sa v tomto období darilo zvyšovať produktivitu vďaka veľkému záujmu zákazníkov nasadzovať nové technológie. Táto dôvera zákazníkov v nové technológie sa neskôr začala prudko znižovať, čo spôsobilo znižovanie ziskov a následné zatváranie menej výkonných softvérových spoločností.

Ako sa softvérový priemysel vyvíja, je čoraz ťažšie zvyšovať produktivitu zlepšovaním technológií. Napriek tomu sa v tejto súvislosti ako zaujímavá oblasť výskumu javí automatické generovanie zdrojového kódu pre dobre definované čiastkové problémy. Príkladom sú generátory objektovo-relačných mapovačov, ktoré odbremeňujú programátora od písania kódu, ktorý transformuje dáta z relačnej databázy do ich objektovej reprezentácie a naopak.

Ľudia a ich pracovný potenciál

Úspech softvérovej spoločnosti závisí najmä na kvalifikovanosti jej zamestnancov. Každý člen softvérového tímu musí mať dostatočné vedomosti a zručnosti na to, aby dokázal efektívne využívať dostupné technológie a vykonávať definované procesy. Úlohou riadiacich pracovníkov je vytvoriť priestor na zlepšovanie vedomostí, zručností, motivácie a výkonnosti členov softvérových tímov. Zvládnuť manažment a

kontinuálny rozvoj pracovnej sily je náročné. Preto vznikol súbor odporúčaní P-CMM (People Capability Maturity Model) [3], ktorý detailne opisuje postup, akým by sa mali realizovať organizačné zmeny. Tie sa musia plánovať dlhodobo, lebo sa na produktivite prejavujú až v dlhšom časovom horizonte.

Všetky metódy na zvyšovanie kvalifikácie pracovníkov pracujú s množinou zamestnancov, ktorí sú v spoločnosti. Na produktivitu preto výrazne vplývajú spôsoby, akým spoločnosť hľadá nových zamestnancov, akým si vyberá medzi uchádzačmi o zamestnanie, ako ich obsadzujú do pracovných pozícií a ako ich presúva na nové pozície. Úspešné prístupy napríklad zahŕňajú ciele ziskavanie nových pracovníkov z radov absolventov a ich profesionálne nasmerovanie na oblasť, v ktorej spoločnosť v budúcnosti predpokladá najväčší nedostatok.

Spôsob obsadzovania nových pracovných miest spravidla vychádza z kultúry danej spoločnosti a z jej dlhodobej stratégie. Častým problémom pri získavaní nových pracovníkov je podľa mňa taká politika organizácie, ktorá nedovoľuje ohodnotiť nových pracovníkov rovnako ako dlhoročných s podobnými vedomosťami. Takýto prístup znižuje motiváciu nových pracovníkov dosahovať vysoké výkony, a teda znižuje potenciálnu produktivitu.

Ďalšou významnou oblasťou manažmentu ľudských zdrojov, ktorá má veľký vplyv na produktivitu pracovníkov, je oblasť *komunikácie a koordinácie*. Jej cieľom je zabezpečiť pravidelnú a efektívnu výmenu informácií v organizácii. Rovnako by mala zabezpečiť, aby každý zamestnanec mal vedomosti a zručnosti nutné na efektívne zdieľanie informácií a na efektívnu koordináciu aktivít s ostatnými členmi tímu. Jednotliví zamestnanci by mali vedieť rozpoznať a predchádzať takým druhom komunikácie, ktoré vedú k vytváraniu nepriateľského prostredia pre iných zamestnancov. Je to napríklad nepriznanie rovnakých šancí na sebarealizáciu, diskriminácia a podobne.

Produktivitu ovplyvňuje aj *fyzické prostredie* v ktorom sa zamestnanec nachádza. Ide napríklad o dostatočné osvetlenie pracoviska či vytvorenie tichého prostredia, ktoré umožňuje pracovníkom lepšie sa sústrediť na svoju prácu. Podľa mojich skúseností je udržiavanie tichého prostredia v softvérových firmách často zanedbávané najmä zo strany manažérov. Tí s cieľom detailne poznať problémy, ktoré riešia vývojári, strávia komunikáciou s nimi veľa času, neuvedomujúc si, že tým rušia aj ostatných nachádzajúcich sa v rovnakej kancelárii.

Na zvyšovanie produktivity je tiež nevyhnutné vybudovať v spoločnosti systém, ktorý umožní pravidelné školenie jednotlivých vývojárov v oblastiach, v ktorých riešia svoje úlohy.

Motivácia pracovníkov

Motivácia je okrem iného aj prevenciou proti problémom s produktivitou a kvalitou práce [1]. Každého zamestnanca motivuje niečo iné. Preto neexistuje jednoduchý návod, ktorý by umožňoval motivovať všetkých zamestnancov. Hľadanie motivačných faktorov, ktoré motivujú jednotlivých zamestnancov, nie je jednoduché. Zamestnanci si

často svoje motivačné faktory ani sami nevedomujú. Keď si ich aj uvedomujú, neradi ich svojim nadriadeným prezrádzajú. Tí často predpokladajú, že ich zamestnancom ide iba o peniaze, a preto sa ani nepokúšajú hľadať iné motivačné faktory. Podľa [4] k najčastejším motivačným faktorom popri mzde patrí uznanie za dobre vykonanú prácu, poskytnutie väčšej dôvery a samostatnosti, širšie vedomie významu práce a umožnenie pružnejšieho časového usporiadania pracovného dňa.

Motivačné metódy rozdelíme na pozitívne a na negatívne. Negatívna motivácia je zvyčajne nejaká forma trestu prichádzajúca na rad v prípade, ak zamestnanec niečo zanedbal alebo nevykonal. Pozitívna motivácia je nejaká forma odmeny za dobre vykonanú prácu. Z dlhodobého hľadiska možno dosiahnuť oveľa vyššiu výkonnosť zamestnancov, keď pred negatívnymi motivačnými metódami uprednostíme pozitívne.

Opakujúcim sa problémom pri pozitívnej motivácii je odmeňovanie rôznych pracovných výkonov rovnakou odmenou. Okrem toho, že je to zjavne nespravodlivé, to v konečnom dôsledku znižuje motiváciu minimálne toho pracovníka, ktorý podal vyšší výkon. Tiež sa tým v organizácii vytvára nepriateľské prostredie, ktoré sa neskôr môže odraziť na celkovej produktivite softvérového tímu.

Zlepšovanie procesu vývoja

Tretí spôsob zvyšovania produktivity softvérového tímu spočíva v zlepšovaní samotného procesu vývoja softvérového systému. Bez využitia dobrého procesu, sa vývoj systému môže stať chaotickým, čo sa následne odrazí v celkovo nízkej produktivite vývoja a v nízkej kvalite systému.

Základnou otázkou, ktorej riešenie je súčasťou definovania procesu vývoja, je kedy sa má vykonávať koordinácia jednotlivých členov tímu. Existujú tri základné prístupy ku koordinácii:

- Big Bang,
- periodická integrácia a synchronizácia,
- integrácia riadená výskytom chýb.

Pre každý projekt musí manažér osobitne zvažovať, aký prístup ku koordinácii členov tímu a k integrácii komponentov použiť. Niekedy sa toto rozhodnutie musí zmeniť aj počas priebehu projektu, a to najmä v prípadoch, keď sa vyskytnú nečakané problémy.

Hlavnou myšlienkou prístupu *Big Bang* je ponechávanie koordinácie a integrácie na koniec projektu. Výsledný produkt sa vytvorí až po dokončení a otestovaní všetkých jeho súčastí. Tento prístup vychádza zo známeho vodopádového modelu. Vývojári nie sú počas trvania projektu zbytočne zdržovaní komunikáciou a môžu sa sústrediť na svoju prácu. Vyžaduje si to však od nich vysokú schopnosť samostatne riešiť vzniknuté problémy. Preto je tento prístup vhodný najmä v tímoch so skúsenými a vysoko motivovanými vývojármi, prípadne pri projektoch s dobre definovanými úlohami. Napriek tomu, že je počas projektu možné zmeniť prístup ku koordinácii za iný, často

je náročné zistiť, kedy nastal ten správny okamih, keď už vieme, že prístup Big Bang prináša viac problémov ako úžitku.

Periodická integrácia a synchronizácia vychádza z iteratívneho modelu vývoja softvéru. Spočíva v definovaní intervalov, v ktorých sa jednotlivé časti systému integrujú. Dĺžka týchto intervalov závisí od druhu projektu, prípadne od toho, ako často potrebujeme poznať aktuálny stav projektu a kvalitu výsledku. Je to najčastejšie používaný prístup ku koordinácii.

Pri takých typoch projektov, s ktorými už tím má určité skúsenosti, je vhodné využiť prístup s integráciou riadenou výskytom chýb. Pri takomto prístupe sa integračné stretnutia zvolávajú až pri výskyte problémov. Nevýhodou je trochu menší prehľad o stave projektu ako pri periodickej integrácii. Ten je však vyvážený znížením množstva neefektívnej komunikácie v tíme.

Záver

Je dôležité zaoberať sa systematickým zvyšovaním produktivity pri vývoji softvérových systémov v tímoch. Dôvodom je ostrá konkurencia medzi softvérovými spoločnosťami, ktorá neumožňuje dlhšie pôsobenie na trhu tým spoločnostiam, ktoré produktivitu neberú vážne.

Tento príspevok sumarizuje dôvody, ktoré vedú organizácie k tomu, aby sledovali vývoj produktivity svojich softvérových tímov a opisuje metódy na priame či nepriame meranie produktivity. Príspevok vymenúva a podrobnejšie rozoberá oblasti, ktoré môže manažment ovplyvniť s cieľom zvýšiť produktivitu.

Použitá literatúra

1. Bieliková, M.: Softvérové inžinierstvo. Bratislava, 2000.
2. Chiang, I. R., Mookerjee, V. S.: Improving Software Team Productivity. In Communications of the ACM, Vol. 47, No. 5 (May 2004), p. 89-93.
3. Curtis, B., Hefley, W., Miller, S.: People Capability Maturity Model. Carnegie Mellon, Pittsburgh, 2001.
4. Chyby pri odmeňovaní a motivácii zamestnancov. *Infoware*, October 2005, p. 19
5. Poppendieck, M.: The software productivity imperative. In Agile Project Management, Vol. 5, No. 3.

Annotation

Improving software team productivity

Productivity is the output of goods and services divided by the inputs needed to generate that output. It is essential that managers in software projects have a fundamental understanding of the methods for significantly improving productivity. In this essay we describe some of these methods using a systematic approach to categorize them.