

SPÔSOB A INTENZITA KOMUNIKÁCIE PRI VÝSKYTE A ODSTRAŇOVANÍ CHÝB

*Bolesť spôsobená disciplínou je oveľa menšia ako
bolesť spôsobená ľutovaním...*

Luboš Masný

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
lubomasny[zavináč]gmail[.]com

Abstrakt. Esej sa zaoberá intenzitou a typom komunikácie pri odstraňovaní chýb. Napriek tomu, že počítače čoraz viac vstupujú do nášho života, poukazuje sa na neustálu dôležitosť osobnej komunikácie. Diskutuje sa, kedy je pri odstraňovaní chýb efektívnejšia priama a kedy nepriama komunikácia. Esej ďalej opisuje spôsob komunikácie, ktorý je vhodnejší v rámci tímu a ktorý pri zákazníkovi. Postupne sa rozoberajú jednotlivé okolnosti tejto vhodnosti. Dôraz sa kladie na závažnosť chyby, na rýchlosť potreby odstránenia, na počet podieľajúcich sa osôb na projekte a ich intenzita práce. Následne je podaný návrh na systém, ktorý by dokázal na základe kritérií vyhodnotiť najvhodnejší spôsob komunikácie.

Kľúčové slová: chyba, komunikácia, odstránenie chyby,

Úvod

Odkedy internet vstúpil do našich životov, pociťujeme úbytok priamej komunikácie. Tak ako nástup internetu, aj tento proces úbytku mal veľmi rýchly spád, hoci to spočiatku bolo len nevinné, občasné posielanie správ. V dnešných dňoch internet na komunikáciu vplýva tak výrazne, že deň bez rozhovoru prostredníctvom počítačov si už ani nevieme predstaviť. Tento trend nezasiahol len širokú verejnosť a uchytil sa vo všetkých oblastiach nášho života. To znamená, že v takmer všetkých firmách a organizáciách komunikujeme prostredníctvom počítačov.

Osobná komunikácia verzus komunikácia prostredníctvom počítačov

Od vzniku softvéru pre počítače hľadáme pre naše potreby kvalitné a bezchybné riešenia. Všetci, od programátorov jednotlivcov až po veľké spoločnosti zamerané na vývoj softvéru, sa snažia priniesť na trh čo najlepšie produkty, ktoré používateľom vyhovujú. Mnoho krát sa však stretávame s nečakaným a nežiadaným výskytom chýb. Preto sa kladie veľký dôraz na to, aby bol proces odstraňovania chýb čo najjednoduchší, zároveň však čo najrýchlejší. Častým problémom výskytu chýb v produkte je zlá komunikácia vo vývojovom tíme, alebo so zákazníkom.

To, že v distribuovaných tímoch je komunikácia prostredníctvom počítačov preferovaná a využívaná, ukazuje štúdia uskutočnená autormi Nadeem, Ebtisam a Muhammad [3], čo je z časových aj finančných dôvodov pochopiteľné. Avšak takáto komunikácia je čoraz viac uprednostňovaná aj v takých prípadoch, kde je zákazník a vývojový tím na jednom mieste. Myslím si, že nie je jednoduché jednoznačne určiť, kedy je lepší osobný rozhovor a kedy komunikácia prostredníctvom počítačov. V mnohých prípadoch je využitie počítača rýchlejšie a efektívnejšie. Pri odstraňovaní chýb je to podľa môjho názoru hlavne v tých prípadoch, keď je opis chyby veľmi jednoduchý a ľahký na pochopenie. V takomto prípade by bolo úplne zbytočné, aby prišiel zákazník chybu referovať osobne, alebo aby sa podieľal na návrhu jej odstránenia. Avšak sa často stáva, že popis chyby nie je jednoduchý. V tomto prípade by bol celý proces vymieňania e-mailov a správ prostredníctvom internetu neprehľadný, časovo veľmi náročný, často krát bez riešenia a nakoniec by aj tak muselo dôjsť k osobnému stretnutiu. Otázkou je, ako teraz rozlíšiť, kedy je potrebné uprednostniť priamu komunikáciu so zákazníkom a kedy nie je nevyhnutná?

Podľa môjho názoru, jedným zo spôsobov, ktorý by mohol napomôcť tomuto rozhodovaniu a určovaniu, aký spôsob komunikácie zvoliť je využívanie systému, ktorý by bol schopný zaznamenávať jednotlivé chyby a ich popis. Tieto systémy sú veľmi jednoduché na používanie aj pre bežných používateľov a umožňujú široko kategorizovať chybu. Jedna z funkcií, ktorú ponúkajú používateľom, je možnosť určenia závažnosti chyby. To znamená, že podľa tohto parametra by sa dalo jednoducho určiť, či je potrebné osobné stretnutie. V prípade, ak zákazníkovi po vykonaní určitého sledu akcií padá celá aplikácia, bol by postup nasledujúci:

Zákazník sa po vytvorení a uložení správy o novej kritickej chybe stretne s vedúcim projektu, aby chybu čo najrýchlejšie odstránili. Stretnutie sa medzi zákazníkom a vedúcim projektu uskutočňuje za predpokladu, že vedúci projektu je ten, čo má najlepší prehľad o celkovom stave projektu. V prípade, že má lepší prehľad iný člen tímu, tak sa stretnutia zúčastní on. K osobnému stretnutiu so zákazníkom by došlo preto, lebo priama komunikácia je intenzívnejšia, flexibilnejšia a kritickej chybu treba v čo najkratšom čase odstrániť. Po spoločnej analýze, za akých okolností vzniká chyba, by vedúci pridelil programátorovi úlohu, aby chybu odstránil.

Avšak čo v prípade, že programátor zistí, že síce chyba spôsobuje pád celej aplikácie, ale riešenie je triviálne a rýchle? Bola nutnosť stretnutia sa so zákazníkom potrebná? Myslím si, že nie. V druhom scenári by mohla byť situácia presne opačná. Zákazník nájde chybu, ktorá nespôsobuje závažné problémy a v systéme chýb jej pridelí nižšiu prioritu. Podľa tohto kritéria by nedošlo k osobnému stretnutiu, vedúci projektu z popisu chyby

určí miesto vzniku a chybu pridelí programátorovi na odstránenie. Avšak čo ak programátor zistí, že za danú chybu nie je zodpovedný a nevie ju opraviť? Oznámi to vedúcemu projektu, ktorý pridelí úlohu inému programátorovi. Situácia, ktorá nastala pri prvom programátorovi sa môže zopakovať aj pri druhom a takto to môže pokračovať aj ďalej. Nakoniec dôjde k vyčerpaniu všetkých programátorov a aj tak bude potrebné zorganizovať stretnutie so zákazníkom a celým tímom. Zákazník bude musieť podrobne opísať čo spravil, keď mu program vyhodil chybu. Vedúci projektu bude musieť spolu s tímom analyzovať, kde chyba mohla vzniknúť a nájsť na jej odstránenie riešenie.

Tieto scenáre nám naznačili, že nebude postačujúce len jedno kritérium, podľa ktorého sa vyhodnocuje potreba osobného stretnutia a bude potrebný ďalší prvok na rozhodovanie. Týmto prvkom by mohlo byť sledovanie pohybu chyby u vývojárov. Takýmto sledovaním by sa zabránilo situáciám, ktoré nastali v predošlých prípadoch.

Ku chybe, ktorá bola označená ako kritická, by sa nevolal okamžite zákazník a naopak, problematická chyba s nižšou prioritou by sa riešila aktívnejším prístupom. Situácie, ktoré môžu nastať, sú nasledujúce:

Aby sa predišlo nepotrebnému stretnutiu so zákazníkom, tak pri kritickej chybe je po analýze vedúcim projektu chyba pridelená programátorovi, ktorý je podľa názoru vedúceho najvhodnejším. Až keď sa tomuto programátorovi nepodarí chybu odstrániť (t.j. chyba nebola pravdepodobne primitívna), tak zorganizuje stretnutie so zákazníkom. Pri chybe s nižšou prioritou, je chyba od vedúceho pridaná postupne viacerým programátorom a až po niekoľkých neúspešných pokusoch sa zavolá zákazník. Za pomoci tohto prvku sa v prípade kritickej chyby predíde zbytočnému volaniu zákazníka, v druhom prípade neskorému zavolaniu. Nechcenou situáciou v oboch prípadoch je zavolanie celého tímu, aj zákazníka, a konečné zistenie, že jedna či druhá strana bola na celom stretnutí úplne zbytočná. Táto situácia môže nastať v takom prípade, že celý problém vyriešil tím vývojárov bez pomoci zákazníka alebo naopak. Čo je ale nutné sledovať, aby sa stretnutia zúčastnili len vhodné strany? Novým pridaným prvkom môže byť sledovanie veľkosti pridávaných častí a počet pracujúcich osôb na jednej časti projektu.

Abreu a Premraj [1] spravili empirický výskum týkajúci sa intenzity komunikácie medzi vývojármi, ktorí sa podieľajú na vývoji vzhľadom na intenzitu výskytu chýb v zdrojovom kóde. Z tohto výskumu prišli k záveru, že čím je intenzita komunikácie vyššia, tým je výskyt chýb pravdepodobnejší. Takýto záver by bolo vhodné a rozumné zakomponovať aj do rozhodovacieho procesu, kedy je potrebné využiť osobnú komunikáciu. Keďže celý proces určovania typu komunikácie už prestáva byť triviálny a sledovanie intenzity komunikácie jednotlivých členov tímu ho robí ešte zložitejším, bolo by užitočné mať na takéto vyhodnocovanie systém. Tento systém by automatizoval všetky doteraz spomínané úkony, pričom by navyše sledoval intenzitu počítačovej komunikácie v tíme. Je dôležité, aby bol do tohto sledovania zahrnutý aj zákazník, keďže býva súčasťou na stretnutiach. Keďže vychádzame z predpokladu, že čím je komunikácia intenzívnejšia, tým je výskyt chýb pravdepodobnejší, je potrebné aby systém označoval časti, na ktorých sa podieľalo viac ľudí, ktorí medzi sebou komunikovali intenzívnejšie. Tieto časti by boli považované za kritické, keďže sú náchylnejšie na výskyt chyby. Preto v prípade, že sa takáto časť v projekte nachádza, je veľmi dôležité aby na to systém upozornil vedúceho projektu. V tomto prípade je riešením ako predísť chybám v rámci tímu, častejšie stretávanie členov, ktoré by pre ostatných slúžilo na rozšírenie prehľadu o práci. Táto

4 *Luboš Masný*

informácia je nesmierne dôležitá pre vedúceho projektu, ako aj pre ostatných programátorov. Ak sa chyba vyskytne napriek týmto opatreniam, systém ponúkne nasledujúce riešenia:

V prípade, že sa na riešení podieľalo viacej programátorov, chyba je kritická a prebehne prvé neúspešné pridelenie chyby, systém navrhne stretnutie tímu. Toto rozhodnutie uprednostní najskôr stretnutie s tímom bráni situácii, kde sa na stretnutí zúčastnia nepotrebné osoby (v tomto prípade zákazník). Až po neúspešnej oprave chyby je na stretnutie pozvaný aj zákazník. V opačnom prípade, že časť programu nie je systémom označená ako kritická na výskyt chyby, ale zákazníkom je chyba označená ako kritická, systém po prvom neúspešnom pridelení chyby programátorovi, navrhne stretnutie so zákazníkom.

Dôvodom je, že vedúci pravdepodobne nemá jasnú predstavu, kde presne sa mohla chyba vyskytnúť a potrebuje bližšiu špecifikáciu. Ak by bola priorita chyby nastavená na nižšiu hodnotu, nechal by sa väčší priestor tímu vývojárov. Najskôr by vedúci skúsil pridelať opravy jednotlivým programátorom, následne by nasledovalo stretnutie tímu a pokusy opráv. Ak by sa nenašlo riešenie, bol by na stretnutie pozvaný aj zákazník.

Podobne, ako pri sledovaní počtu zúčastnených osôb, je vhodné sledovanie veľkosti pridávania nových častí do projektu. Čím bude veľkosť nových častí menšia, tým väčšia bude snaha riešiť problém v rámci tímu. Malé časti kúsokujú projekt, a tak môže nastať vzájomná nekonzistencia s časťami, ktoré pridali ostatní programátori. V prípade, že by sa pridávalo menej častí s väčším rozsahom, bude pri riešení problému uprednostnené stretnutie so zákazníkom.

Dôvod je prostý. Je ľahšie otestovať spoluprácu pár veľkých častí programu (samozrejme za predpokladu, že samostatne pracujú bez akejkoľvek chyby) a teda je pravdepodobnejšie, že chyba nenastala v tejto preskúmanej časti programu. Teda je potrebné nájsť nové miesto možného výskytu chyby – za spolupráce so zákazníkom.

Avšak ani takýto sledovací systém nemôže nahradiť ľudskú intuíciu. Vedúci projektu sa môže presne riadiť pokynmi systému, aby bolo jeho rozhodovanie čo najefektívnejšie a aby nezaťahoval do problému osoby, o ktorých pomoci pri riešení pochybuje.

Niekedy všetky spôsoby zlyhávajú a je nutné počúvať vlastnú intuíciu. Takéto situácie môžu nastať kedykoľvek. Ako príklad môže slúžiť pridanie úlohy programátorovi prostredníctvom počítača. Je pravda, že ak sa na pridanie úlohy použije nepriama komunikácia, má vedúci spätnú väzbu od programátora zdokumentovanú. A taktiež je pravda, že zodpovednosť za túto chybu spadá na samotného vývojára, ktorý potvrdil, že tú chybu je schopný opraviť. Avšak čo v takom prípade, keď ju nakoniec vývojár nie je schopný opraviť? Alebo čo keď ju nestihne opraviť?

Áno, síce je celá zodpovednosť na programátorovi, ale nájdenie vinníka ešte nie je riešením problému v programe. Následky v prípade zlyhania riešenia môžu byť katastrofálne a je jedno kto to má na zodpovednosti. Problém môže znížiť zisk, spôsobiť stratu projektu (a tým by šiel projekt do mínusu), ba dokonca môže spôsobiť krach spoločnosti.

Samozrejme, že jedna malá chyba pravdepodobne nenapácha takéto obrovské škody, ale ak by sa do interakcie dostalo viacero vážnych chýb, takéto následky môžu byť reálne. K tomu je niekedy riešenie úplne jednoduché – nesnažiť sa byť čo najefektívnejší a najohľaduplivejší pre všetky strany a použiť starú osvedčenú metódu osobnej

komunikácie. Pri priamej komunikácii sa môže vedúci projektu uistiť, že chyba je správne pridelená, pomocou vnímania reči tela programátora. Istotou a sebadôverou v hlase môže rozpoznať, či je danú chybu schopný opraviť alebo nie. Ak má vážnejšie pochybnosti, môže prideliť úlohu niekomu inému. Vďaka tomu, že je komunikácia osobná, je možné predísť spomínaným katastrofickým scenárom.

Záver

Pri odstraňovaní chýb je efektívna priama, ale aj nepriama komunikácia. Väčšinou sa ku žiadanému výsledku dostaneme obidvoma spôsobmi. Avšak, keďže je dnešný svet založený na termínoch, je veľmi dôležité, aby bola zvolená komunikácia rýchla a efektívna. Pri rozhodovaní, aký typ komunikácie je vhodnejšie použiť, dokáže byť nápomocný navrhovaný systém na podporu určovania spôsobu komunikácie. Ten vyhodnocuje najvhodnejší spôsob komunikácie pomocou sledovania zadaných parametrov, na ktoré sa teda nemusí sústrediť vedúci projektu. Napriek tomu, že systém vyhodnotí aká komunikácia býva za daných okolností najefektívnejšia, konečné slovo má vedúci projektu. A ako sme ukázali na vzorových situáciách, niekedy, aby sme dosiahli lepšie riešenie, musíme spraviť aj nepredvídateľné kroky.

Použitá literatúra

1. Abreu, R., Premraj, R.: *How Developer Communication Frequency Relates to Bug Introducing Changes*, In Proceedings of the joint international and annual ERCIM workshops on Principles of software evolution (IWPSE) and software evolution (Evol) workshops, 2009
2. Hall, T., Wilson, D., Rainer, A., Jagielska, D.: *Communication: The Neglected Technical Skill?* Published in The global information technology workforce, 2007
3. Dr. Nadeem, E., Ebtisam, M., Muhammad, A.: *Impact of Computer-Mediated Communication on Virtual Teams' Performance: An Empirical Study*. ACM Digital Library

Annotation

Type and intensity of communication in debug process

The whole essay is about communication. It's about advantages and disadvantages of personal and computer mediated communication. What type of communication is better for development team and what kind of communication for customer. At the beginning are mentioned important facts, which can decide type of communication. When it's too much complicated, there is a suggestion about having system. System is tracking all the fact and making decisions about next step in communication.