

Memex po šesťdesiatich rokoch

STANISLAV OCHOTNICKÝ

*Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
sochotnický[zavínáč]kmit[.]sk*

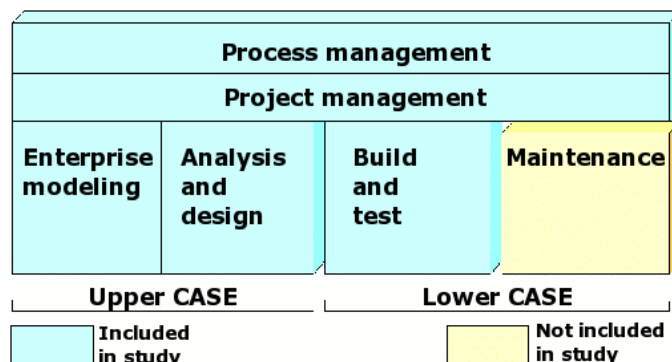
Abstrakt. Zahľtenie informáciami tiež nazývané informačné znečistenie je problém dotýkajúci sa mnohých oblastí ľudskej činnosti. Súvisí s nástupom Internetu v posledných dvadsiatich rokoch a rozšíreným využívaním rôznych komunikačných prostriedkov. Táto práca sa zaoberá využívaním podporných prostriedkov na manažment softvérových projektov, plánovanie, manažment ľudských zdrojov a ich súvis s informačným zahľtením. Poskytuje tiež pohľad na podporné nástroje pre manažment softvérových procesov a prístupy k ich používaniu, ktoré môžu problémy s informačným zahľtením minimalizovať. V rámci toho predstavuje aj zamyslenie sa nad vplyvom používania CASE nástrojov na úspešnosť, cenu a ďalšie aspekty tvorby softvérových systémov. Na záver je zvlášť venovaná pozornosť rozdielom v menších tímoch, v ktorých sa roly jednotlivých členov tímu menia v čase, ako je to napríklad v tímových softvérových projektoch na univerzitách.

Práve začíname, alebo prečo sme ešte nezačali

Ľudia využívali stroje na zamaskovanie vlastných nedostatkov od nepamäti. Začalo to pästným klinom a aspoň zatiaľ skončilo u počítačov a sondami vo vesmíre. Už v roku 1960 si však v L.C.R Licklider vo svojej eseji Man-Computer Symbiosis [7] uvedomil jednostrannosť takýchto vylepšení. Takmer vždy sa jednalo o vylepšenie fyzických nedostatkov, ktorých nám príroda dala neúrekom. Naše rozumové schopnosti však stáli bokom. Jeden návrh, ktorý sa pokúsil túto nevyváženosť čiastočne vyrovnať bol spomenutý v práci As We May Think od V. Busha a to ešte v roku 1945 [2]. Ako riešenie v nej V. Bush navrhol zariadenie nazvané Memex. Cieľom tohto zariadenia malo byť zaznamenávanie a efektívne zdieľanie myšlienok, myšlienkových postupov a iných informácií. V. Bush je vďaka tomuto návrhu často považovaný za otca myšlienky hypertextu. Okrem iného však V. Bush postrehol aj prudký nárast množstva informácií, ktoré si našli cestu do života každého z nás: „(Man) has built a civilization so complex that he needs to mechanize his records more fully if he is to push his experiment to its logical conclusion and not merely become bogged down part way there by overtaxing his limited memory“. Bol teda aj medzi tými, ktorí si uvedomovali,

že niekedy je menej viac. Problémy vyplývajúce z množstva informácií boli napokon súhrnne pomenované zahltenie informáciami [8,10]. Nejedná sa iba o samotné množstvo informácií, ale aj o neschopnosť porozumieť informáciám, alebo neschopnosť zhodnotiť či informácia existuje. V posledných desaťročiach nie je teda problém s neexistenciou informácií, ale vznikol problém s ich efektívnym vyhľadávaním a hodnotením. Nie sme teda o veľa bližšie k ideálu jednoduchého zdieľania informácií ako pred šesťdesiatimi rokmi, v mnohých ohľadoch sme na tom dokonca o mnoho horšie.

Možno by ste sa mohli spýtať čo má hypertext či nebodaj Memex spoločné s vývojom softvéru. Nuž, tak ako väčšina ľudských činností, aj vývoj softvéru sa často odohráva v tímoch zložených z ľudí. Koordinácia, riadenie, pridelovanie úloh a efektivita ich riešenia v tíme nesporne závisí aj od informácií, ktoré majú jednotliví členovia k dispozícii. Spôsob vytvárania a distribúcie informácií v softvérových tímoch je možné podporiť rôznymi softvérovými prostriedkami. S rastúcou komplexnosťou softvérových projektov sa využitie podporných softvérových prostriedkov stalo nutnosťou. Efektívne prostriedky na výmenu informácií v softvérových tímoch však stále nie sú plne rozvinuté a tzv. „silver bullet“ čiže všeobecne použiteľné riešenie sa stále nenašlo.



Obr. 1. Nástroje zahrnuté v štúdiu o využívaní CASE nástrojov (prebrané z [5]).

Menej je niekedy viac

Manažment softvérových procesov je súčasťou všetkých vývojových štádií, od inicializácie projektu až po jeho ukončenie. V rôznych fázach vývoja je však potrebné pristupovať k manažmentu ľudí a úloh odlišne. Preto aj používané podporné prostriedky sa prispôbujú jednotlivým fázam vývoja. Použité nástroje by mali zmenšiť režijné náklady, skvalitniť výsledky jednotlivých etáp vývoja, ale aj podporovať kladné vnímanie procesu manažmentu. Práve od vnímania vplyvu používania podporných prostriedkov závisí ich ďalšie použitie.

Tento empirický poznatok bol potvrdený v štúdií zaoberajúcej sa využívaním CASE nástrojov vo vyše stovke projektov [5]. Skúmané boli nástroje používané v rámci všetkých fáz vývoja okrem údržby softvéru (pozri Obr. 1). V štúdií boli delené CASE nástroje na takzvané upper CASE a lower CASE. Upper CASE nástroje sú používané vo fáze špecifikácie a návrhu. Lower CASE potom vo fáze vývoja a testovania. Počas štúdie sa okrem iných zistili nasledovné skutočnosti:

- používanie lower CASE nástrojov zvyšuje pravdepodobnosť, že sa projekt oneskorí,
- tímy s horšou koordináciou zvykli používať CASE nástroje vo väčšej miere,
- využívanie upper CASE nástrojov spôsobilo nižšiu kvalitu návrhových dokumentov, ale investori (respektíve zákazníci) boli spokojnejší ak sa tieto nástroje používali,
- využívanie CASE nástrojov v menších projektoch zvykne byť nižšie. Taktiež majú nižšiu cenu práce a spokojnosť používateľov systému,
- počiatočné školenia o používaní CASE nástrojov vyústili do väčšej spokojnosti medzi členmi tímu.

Vo výsledku bolo zhodnotené, že použitie CASE prostriedkov a prostriedkov na podporu manažmentu pôsobilo ako lupa. V prípade, že členovia tímu neboli schopní používať formálne metódy na výmenu informácií, boli ich výsledky pri použití CASE nástrojov výrazne horšie ako keď CASE nástroje nepoužili. Naopak v prípade tímov kde boli členovia školení v používaní CASE nástrojov a formálnych metód návrhu došlo k zvýšeniu výkonu a vyššej spokojnosti odberateľov. Faktory, ktoré zjavne vplývali na zistené výsledky, boli okrem iného slabá úroveň komunikácie v tíme a z toho vyplývajúca nedostatočná znalosť používaných CASE nástrojov. Do úvahy treba isto brať aj to, že štúdia prebiehala v rokoch 1993-1997 kedy úroveň CASE nástrojov nebola na úrovni dnešných asistenčných nástrojov.

Jedným z najkontroverznejších záverov celej štúdie prekvapil aj samotných autorov. Zníženie kvality návrhových dokumentov pri využití CASE nástrojov bolo nečakané. Empiricky sa predpokladalo, že využívanie pomocných nástrojov uľahčí a zefektívni komunikáciu v tíme a tým skvalitní výstupy. Práve tu je možné nájsť paralelu medzi vývojom softvéru a informačným zahltením. Návrhári mali k dispozícii nástroje, ale ich použitie si vyžadovalo informácie, ktoré nemali alebo neboli ochotní ich získať. Tento predpoklad potvrdzuje aj novšia štúdia [6]. Táto štúdia prišla aj s novým pohľadom na informačné znečistenie. Vývojári sa súčasne sťažovali na dva problémy, ktoré im bránili v efektívnejšej práci:

- nemám dostatok potrebných informácií, aby som mohol robiť svoju prácu,
- mám príliš veľa informácií.

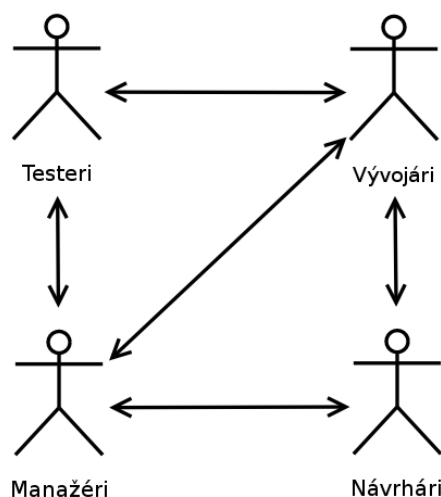
Tieto dve zdanlivo protikladné tvrdenia iba potvrdzujú potrebu lepšej správy informačných tokov pri vývoji softvéru. Otázkou ostáva, ako môžeme vylepšiť komunikáciu, manažment znalostí, ľudských zdrojov a úloh v tíme.

Pravidlá a dobrovoľný prístup k informáciám

Ako už bolo spomenuté, myšlienkový pochod „viac informácií = lepšia informovanosť“ už bol dávno prekonaný. Ako teda môžeme pomôcť členom tímu, aby boli informovaní o dianí na pracovisku (workspace awareness, pozri [11]) a pritom ich nezahľcovať nepotrebnými informáciami?

Najprv potrebujeme identifikovať jednotlivé podskupiny ľudí v softvérovom tíme:

- manažéri,
- návrhári,
- vývojári,
- testeri.



Obr. 2. Zjednodušené komunikačné cesty medzi členmi tímu pri vývoji softvérového systému.

Komunikácia každý s každým by isto priniesla nemalé problémy a spôsobila informačné zahltenie u mnohých členov tímu. Komunikačné cesty závisia často od vývojového modelu, zloženia tímu a ich umiestnenia v priestore [12]. Zjednodušený všeobecný model by mohol vyzerat' podobne ako Obr. 2. Je zjavné, že najviac vyt'azenými členmi tímu v tomto ohľade sú manažéri a vývojári. Z obrázku nie je zrejmé, že vývojári síce komunikujú s návrhármi aj testermi, no zvyčajne v odlišných fázach vývoja. Navyše každá podskupina tímu používa na riešenie problémov inú sadu nástrojov, ktoré nie je vždy možné prepojiť.

Podporné nástroje, s pomocou ktorých si členovia tímu vzájomne odovzdávajú informácie, je viacero:

- elektronická pošta,
- bázy poznatkov,
- systémy sledovania chýb,
- systémy na manažment ľudských zdrojov a úloh,
- komunikácia v reálnom čase (instant messaging/IM).

Elektronická pošta

Elektronická pošta je jednoznačne najpoužívanější systém na komunikáciu. Jeho najväčšou výhodou je univerzálnosť a rýchlosť s ktorou je možné distribuovať informácie. Bohužiaľ spolu s IM je jedným z najvýraznejších faktorov spôsobujúcich informačné zahltenie. Pre efektívne používanie, ktoré zníži informačné zahltenie navrhol Jakob Nielsen [9] nasledovný spôsob použitia elektronickej pošty pre jednotlivcov:

- nekontrolujte si poštu príliš často,
- nepoužívajte „odpovedať všetkým“, ale vyberajte si komu správu pošlete,
- používajte informatívne predmety správ,
- vytvorte si zvláštny schránku na osobné správy, alebo používajte triedenie,
- píšite krátko a výstižne,
- vyhýbajte sa IM.

Tieto pravidlá je možné s miernymi úpravami použiť aj pri iných formách elektronickej komunikácie v tíme.

Bázy poznatkov

Bázy poznatkov predstavujú stály zdroj informácií v tíme. Mali by slúžiť predovšetkým na zdieľanie informácií, ktoré presahujú rámec jedného projektu. Ide napríklad o firemné know-how, rôzne tipy na riešenie bežných problémov, pravidlá a štandardy používané v rámci tímu atď. Jednotliví členovia tímu si môžu nájsť informácie vždy keď ich potrebujú vďaka podpore vyhľadávania. Jeden z moderných spôsobov udržiavania bázy poznatkov je využitie kolaboratívneho systému, do ktorého môžu prispievať všetci členovia tímu. Tento spôsob prináša ale ďalšie problémy týkajúce sa zodpovednosti za udržiavanie aktuálnosti informácií.

Výhodou takéhoto uchovávaní informácií, ktorá sa prejaví pri dlhších projektoch je najmä nezávislosť informácií na príchode a odchode členov tímu.

Systémy sledovania chýb

Systémy sledovania chýb (bug-tracking systémy) sa používajú najmä v implementačnej a testovacej fáze vývoja. Umožňujú ľahkú distribúciu a sledovanie úloh v kontexte zdrojových súborov a jednotlivých zmien v kóde. V spojení so systémami na verziovanie umožňujú často sledovanie a monitorovanie práce členov tímu. Práve možnosti prepojenia a uľahčenia práce projektových manažérov pri sledovaní zmien v projekte sú ich najsilnejšou stránkou. Ich veľkou výhodou je aj úplná adresnosť informácií. Členovia tímu majú informácie o časti systému, ktorú majú na starosti a informácie o zvyšných častiach ich nezaťažujú.

Systémy na manažment ľudských zdrojov a úloh

Tieto podporné prostriedky uľahčujú plánovanie a monitorovanie postupu projektu v širšom kontexte. Používajú sa v priebehu celého projektu. Bohužiaľ to tiež často znamená, že nie je možné k úlohe prideliť potrebné množstvo detailných informácií. Na najnižšej úrovni riadenia, teda v tímoch s približnou veľkosťou do 10 ľudí, už tieto systémy strácajú účinnosť v distribuovaní informácií, stále však ostáva funkcionálnosť sledovania postupu projektu a plánovania. Tieto systémy bývajú tiež prepojené na celopodnikový informačný systém, prípadne groupware a pomáhajú aj v strategickom plánovaní.

Komunikácia v reálnom čase

V tomto prípade sa jedná o známe služby typu icq, alebo jabber. Na prvý pohľad veľmi vhodný spôsob na adresnú výmenu aktuálnych informácií. Nesie však so sebou veľmi vysokú cenu v neustálom narušení sústredenia. Existujú informácie o tom, že 1 minúta prerušenia činnosti stojí pracovníka 10-15 minút času, ktorý venuje opätovnému návratu k práci, ktorú robil [8]. Samozrejme aj IM má svoje miesto v tímovej komunikácii. Mal by sa však využívať s mierou a malo by byť určené, v ktorých prípadoch je povolené kontaktovať člena tímu v reálnom čase. Väčšinou sa zrejme bude jednať o kritické situácie vyžadujúce okamžitú reakciu.

Zvláštnou kategóriou je samozrejme osobná komunikácia. Tá síce tiež môže spôsobovať problémy so sústredením, no je nevyhnutná z rôznych sociálnych dôvodov. Niektoré výskumy tiež ukazujú, že v menších tímoch umiestnených na jednom mieste sa zvyšuje výkon tímu a kvalita projektu [4,12].

Budúcnosť na dohľad

Existuje veľké množstvo nástrojov na podporu vývoja v softvérových projektoch. Zjednodušene by sme ich mohli rozdeliť do dvoch skupín:

- všeobecné nástroje, ktoré majú problém nadviazať na konkrétny kontext potrebný na riešenie daného problému,

- špecializované nástroje na riešenie jednej triedy problémov.

V poslednom desaťročí sa objavil nový prístup k prezentácii informácií používateľom. Jedná sa o takzvaný task-based (context-sensitive) user interface, teda úlohovo orientované používateľské rozhranie [1]. Je založený na predpoklade, že používateľ chce mať k dispozícii široké spektrum funkcií v rámci jedného nástroja, ale väčšinou potrebuje iba malú časť celkovej funkcionality. To umožňuje znížiť riziko informačného zahltenia a zefektívňuje prácu v tíme. Samozrejme tento prístup vyžaduje aby človek, ktorý zadáva úlohy vedel posúdiť kontext, v ktorom bude musieť jeho tímový kolega pracovať. Úlohovo orientované rozhranie tiež nesmie obmedziť používateľa v prístupe ku kompletným informáciám v prípade potreby.

Asi najznámejším projektom využívajúcim tento prístup vo väčšej miere je Eclipse [3]. Táto jeho črta sa dá ešte posilniť napríklad rozšírením Mylyn, ktoré je možné napojiť na centrálnu správu úloh a takýmto spôsobom pristupovať jednotným spôsobom k úlohám od počiatočného štádia projektu.

Problém škálovateľnosti

Štúdie a väčšina odkazov v tomto článku pojednáva o tímoch, ktoré majú často niekoľko desiatok až stoviek ľudí. Je len prirodzené, že spôsob riadenia a výmeny informácií v menších tímoch sa bude riadiť čiastočne inými pravidlami. Tieto menšie tímy sa často vyskytujú v akademických podmienkach na tímových projektoch.

Pravdou je, že väčšina pravidiel, ktoré platia pre veľké tímy, platia aj pre malé. Pri nesprávnom používaní podporných prostriedkov môže veľmi jednoducho dôjsť k informačnému zahlteniu niektorých členov tímu. Výrazným faktorom, ktorý zväčšuje riziko je aj edukačný charakter tímových projektov. Členovia tímu často „rotujú“ v jednotlivých pozíciách, a tak sú neustále vystavovaní novým pohľadom na projekt. Spolu s týmito pohľadmi musia používať väčšie množstvo nástrojov ako je zvykom vo väčších tímoch so statickým pridelením rolí. Majú síce možnosť overiť si väčšie množstvo poznatkov, no za cenu zhoršenej kvality výstupov projektu a väčšieho informačného zaťaženia.

Vzhľadom k veľkosti tímov sa však zvlášť ponúka využívanie niektorých komunikačných prostriedkov. Jedná sa napríklad o kolaboratívne prostredia na zdieľanie informácií ako bázy znalostí. V menších tímoch je jednoduchšie ustrážiť aktuálnosť jednotlivých informácií a teda nevznikajú problémy dotýkajúce sa väčších tímov. V týchto tímoch taktiež často platí, že nie je problém stretnúť sa osobne na niekoľko hodín a mnoho problémov sa dá riešiť najefektívnejšie práve takýmto spôsobom.

Ďalším aspektom, ktorý je vhodné brať do úvahy je pravdepodobnosť využívania nástrojov, ktoré už aspoň jeden z členov tímu pozná. Rozhodnutia robia priamo členovia tímu a môžu tak priamo ovplyvniť aké nástroje budú používať. Toto najmä vo väčších tímoch nemusí byť vždy pravda.

Zhrnutie

Cieľom tejto práce bolo načrtnúť problémy, ktoré môže spôsobovať informačné zahltenie členov tímu pri vývoji softvéru. V náväznosti na úvod by bolo možno vhodné pýtať sa do akej miery sa v riešeníach blížime k ideálu jednoduchého zdieľania informácií, ktorý by ponúkal Memex. Tu nám neostáva iné len konštatovať, že doterajšia technika ešte do tejto úrovne nedospela. Nesporne bude potrebný ďalší výskum vo viacerých vedeckých odboroch, najmä v odboroch súvisiacich s interakciou človeka a počítača. Je možné, že návrh nástrojov, ktoré zjednodušia proces tvorby softvéru bude vyriešený v rámci všeobecnejšieho výskumu zameraného na psychológiu, kogníciu a príbuzné vedy. Výskum v oblasti informačného zahltenia sa zameriava prevažne na všeobecné informačné systémy a e-mail. Bolo by ale vhodné vykonať štúdie zamerané priamo na spôsob akým informačné zahltenie ovplyvňuje prácu na softvérových projektoch v tímoch používajúcich rôzne podporné nástroje.

Podstatným prvkom v riadení projektov, sú ale bezpochyby ľudia. Preto aj najdôležitejším vybavením softvérového tímu nie sú nástroje, ale štandardy a pravidlá, ktorými sa budú členovia tímu riadiť. Pokiaľ bude mať tím k dispozícii najnovšie nástroje a nebude s nimi vedieť efektívne pracovať, je na tom horšie, ako keby tieto nástroje nemal vôbec.

Použitá literatúra

1. Bomsdorf B., Szwillus G.: Tool Support for Task-Based User Interface Design – A CHI'99 workshop. ACM SIGCHI Bulletin, Vol. 31, No. 4 (1999), 27-29
2. Bush V.: As We May Think. Atlantic Monthly, júl 1945.
3. Eclipse Foundation: <http://www.eclipse.org>, október 2007.
4. French A., Layzell P.: A Study of Communication and Cooperation in Distributed Software Project Teams. In: *Proc. of International Conference on Software Maintenance*, (1998), 146-154.
5. Guinan J.P., Coopridge G. J., Sawyer S.: The Effective Use of Automated Application Development Tools. IBM Systems Journal, Vol. 36, No. 1. (1997), 124-139.
6. Heike F.: The Impact of Computer Mediated Communication on Information Overload in Distributed Teams. In: *Proc. of the 32nd Hawaii International Conference on System Sciences* (1999).
7. Licklider L.C.R.: Man-Computer Symbiosis. In: *IRE Transactions on Human Factors in Electronics* (1960), 4-11.
8. Nielsen J.: IM, Not IP (Information Pollution), ACM Queue, Vol. 1, No. 8, november 2003.

9. Nielsen J.: Ten Steps for Cleaning Up Information Pollution. Jakob Nielsen's Alerbox (január 2004).
10. Nelson R.M.: We Have the Information You Want, But Getting It Will Cost You: Being Held Hostage by Information Overload. ACM Crossroads, Vol. 1, No. 1 (september 1994), 11-15.
11. Sarma A., Hoek van der A., Redmiles F.D.: A Comprehensive Evaluation of Workspace Awareness in Software Configuration Management Systems. In: *IEEE Symposium on Visual Languages and Human-Centric Computing* (2007), 23-26.
12. Teasly D.S., et.al.: Rapid Software Development Through Team Collocation. In: *IEEE Transactions on Software Engineering*, Vol. 28, No. 7 (júl 2002).

Annotation

Memex after sixty years

Problem of information overload, also called information pollution is interfering with whole areas of human activity. It is related to growing use of Internet in last twenty years and increase of communication tools numbers. This essay is dealing with usage of support tools for management of software projects, planning, management of human resources and their connection to information overload. It also offers view of support tools for management of software processes and methods of their use, that can minimize problems with information overload. Part of this view is a speculation on the impact of CASE tool usage on success rate, price and another aspects of software systems development. Particular attention is dedicated to differences emerging in smaller software development teams, where team members often change their roles as time goes on, such as team projects at universities.