

Struktura programátorského týmu

Petr Píkrýl

1 Úvod

.REM [Mayr90 – p.731]

Jedním z poslání managementu v oblasti softwarového inženýrství (Software Engineering) je efektivní řízení vývoje programového vybavení (software). Cílem je zorganizovat lidské zdroje a technické prostředky tak, aby byly optimálně využity. Tento příspěvek se bude zabývat především lidmi zapojenými do procesu vývoje software. Bude se zabývat týmy pracujícími na nejnižším stupni organizační hierarchie, kde počet členů týmu nepřekračuje hodnotu 15 až 20.

Postupy práce, které se při vývoji programového vybavení používají (strukturované programování, modularizace, používání standardních rozhraní a podobně), umožňují rozdělit práci na řešeném projektu na menší, lépe zvládnutelné celky. Strukturované rozčlenění lidské práce má následující výhody:

- Snižuje se složitost. Redukce složitosti problému přímo vede k redukci složitosti organizačních struktur, které jsou potřebné k jeho řešení.
- Úloha se dělí na uzavřené (ucelené) části, které lze přidělit členům týmu. Současně s přidělením prací na části projektu je řešiteli přidělena i zodpovědnost za její vyřešení, což umožňuje lepší dohled nad celkovým průběhem vývoje projektu.
- Správné rozdělení práce snižuje časové nároky na komunikaci mezi členy týmu. Odpovídá-li složitost úkolu schopnostem řešitele, snižuje se čas potřebný na vyjasňování problémů. Tento čas může být věnován produktivní práci.
- Specializace je ekonomičtější. Schopnosti většiny lidí v procesu vývoje programového vybavení nejsou stejné, liší se podle typu úkolu. Přidělení úkolu členu týmu, který se na danou problematiku specializuje, vede k dosažení vyšší efektivity i kvality.

2 Vztahy programátorský tým – nadřízená složka

Programátorský tým musí zapadnout do celkové (nadřízené) organizační struktury. Vztah nadřízených složek ke svým zaměstnancům (zde členům programátorského týmu) ovlivňuje jejich psychiku a tím i jejich pracovní výkon. Organizace může svůj vztah

k zaměstnanci dávat najevo různým způsobem. Hlavním cílem je motivace k práci. Pohled na zaměstnance musí odhalovat, jakou formu a jakou míru motivace potřebuje k optimálnímu výkonu. Způsob odměňování daného člena usměrňuje jeho činnost, definuje jeho roli. V praxi vystupují do popředí následující pohledy zaměstnavatele na své zaměstnance:

- racionálně-ekonomický pohled
- sociální pohled
- pohled z hlediska seberealizace jedince

1. Racionálně ekonomický pohled předpokládá, že primární motivací k práci jsou peníze. Množství peněz, které zaměstnanec za svou práci dostává, může dokonce vyvolávat nebo tlumit emoce a ovlivňovat mezilidské vztahy. Tomuto způsobu řízení je vytýkán jistý stupeň nelidskosti. Technické zájmy převládají nad zájmy lidskými. Vše je velmi organizované, řízení je centralizované. Vytvářejí se různá regulační opatření, narůstá počet byrokratických úkonů. Organizační struktura se začíná dělit do mnoha úrovní, každá z těchto úrovní má jen úzký a přesně definovaný rozsah činností [Knud79].

.REM [Mayr90-p.735]

2. V jiných organizacích dávají přednost sociálnímu pohledu. Organizace se liší od předchozího typu ve svých cílech, způsobu řízení, struktuře a způsobu odměňování.

Organizace tohoto typu dává svým zaměstnancům najevo, že je orientovaná více na zaměstnance než na technické cíle. Snadněji potom může prosazovat své cíle. Vnitřní komunikace je ve srovnání s předchozím typem organizace méně formální. Řízení se přesouvá z centra do skupin, rovněž stimulační a odměny nejsou směřovány na jednotlivce, ale na skupinu, navíc nemají jen čistě finanční podobu [Knud79].

3. Vyšší stupeň sociálně orientovaných organizací vychází z poznání, že nejvyšší úroveň potřeb jednotlivce má svůj základ ve snaze o seberealizaci. Takové organizace se zabývají i vývojem svých zaměstnanců. Manažer organizace definuje problémy a úkoly jen rámcově. Plán a rozhodnutí, jak daný problém řešit, leží na ostatních členech organizace. Organizační struktura se přizpůsobuje problémům, které se právě řeší. Tento přístup vede k tomu, že si jedinci v organizaci postupně sami hledají svá místa. Odměny členové organizace nalézají v práci samotné. Ve spojení s předchozím faktem to znamená, že si každý hledá svůj způsob odměny. Hlavním úkolem manažera je vytvářet prostředí, které takový způsob práce umožňuje [Knud79].

Z tohoto pohledu by se mohlo zdát, že posledně zmíněný přístup je nejperspektivnější a tedy nejvýhodnější, protože touha po seberealizaci bývá stavěna na nejvyšší stupínek žebříčku hodnot člověka. Ale pokud má být motivace k práci založena na snaze o seberealizaci, musí být naplněny potřeby z nižších stupňů žebříčku hodnot člověka. Mimo jiné to znamená, že člen organizace musí být dostatečně finančně zajištěn. Musí mít pocit, že

peníze už pro něho nejsou tím nejdůležitějším. To je však do značné míry i otázka psychická. Obecně platí: jakmile je nějaká potřeba naplněna, ztrácí motivační sílu.

Různí lidé se však nacházejí na různém stupni vývoje. Není neobvyklé, že dosud nenaplnili své potřeby ležící na nižších stupních žebříčku hodnot. Například mladí lidé, vstupující do pracovního procesu, potřebují získat nezávislost, tedy i dostatek prostředků (finančních) pro život. Dalším důvodem toho, že posledně jmenovaný přístup nemusí být vždy nejvýhodnější, je skutečnost, že touha po seberealizaci nemusí být zaměřena na zaměstnání. Někteří vidí svůj vrchol například v rodinném životě, v nějakém koníčku, v amatérském sportu nebo v jiné oblasti, která se zaměstnáním přímo nespojuje.

Závěr, který vyplývá z těchto skutečností, je následující: organizační struktura, způsoby vedení a způsoby odměňování musí být natolik pružné, aby se daly přizpůsobit každému jednotlivci. Ideálním stavem je syntéza všech tří přístupů.

.REM [Mayr90-p.752]

3 Tým vedoucího programátora (The Chief Programmer Team – CPT)

Jedna z prvních zveřejněných týmových koncepcí je známa pod označením *tým vedoucího programátora* (také *tým hlavního programátora* – the Chief Programmer Team, dále CPT [Mill72], [Mill83]). Některé týmy byly při použití této koncepce úspěšné vždy, jiné týmy nedokázaly počáteční úspěchy zopakovat. Ukázalo se, že jedním z důvodů neúspěchu byl nedostatek odborných schopností. Sebelepší koncepce nemůže tento nedostatek odborných schopností vykompenzovat. Dalším z důvodů neúspěchu byla skutečnost, že ne každý člen týmové struktury splňoval požadavky na něho kladené. Týmová koncepce musí být vyladěna podle členů týmu a podle nadřazené organizační struktury, pod kterou tým spadá. Nikoho asi nepřekvapí skutečnost, že má-li být zvolená forma pro určený scénář prací na projektu co nejúčinnější, musí být základní koncepce přizpůsobena situaci.

Jiné týmové koncepce, jejichž stručný popis naleznete v tomto příspěvku, se označují jako *chirurgický tým* (the Surgical Team, dále ST [Broo75]), *revidovaný tým vedoucího programátora* (the Revised Chief Programmer Team, dále RCPT [Mccl81]) a *programátorský tým pracující s potlačením osobnosti* (the Egoless Programming Team, dále EP [Wein71]).

3.1 Hlavní rysy

Cílem této struktury programátorského týmu je poskytnout formální definici vztahů uvnitř týmu, definovat jasně způsob vedení týmu a vytvořit podmínky pro funkční specializaci. Explicitně je definován i způsob výměny informací, definovány jsou vztahy mezi všemi členy týmu a jejich pracovním zařazením. Přísná hierarchická struktura usnadňuje požadovaný centrální charakter řízení, zjednodušuje kontrolu práce jednotlivých členů

a umožňuje udržovat přehled o stavu vyvíjeného produktu. Nejméně dva členové týmu, vedoucí programátor a druhý programátor (backup programmer), dokonale znají každý problém týkající se projektu. Tím je zaručena návaznost vývoje projektu v případě změny personálního obsazení.

3.2 Složení týmu

Jádro týmu je tvořeno následujícími jednotlivci (jim přísluší uvedená odpovědnost):

1. **Vedoucí programátor (the chief programmer – CP).** Vedoucí programátor je jedním z technických expertů týmu a jeho postavení mu přináší následující povinnosti:

- je odpovědný za úspěch týmu;
- zpracovává všechny dokumenty týkající se počáteční fáze projektu (požadavky, specifikace, návrh);
- programuje a testuje kritické části systému;
- dohlíží na práci ostatních členů týmu, určuje směr jejich práce.

Z těchto důvodů musí mít vedoucí programátor velmi dobré technické znalosti a nejméně dobré schopnosti manažerské.

2. **Druhý programátor (the backup programmer – BP).** Druhému programátorovi je přidělena role záložního vedoucího. Musí být dokonale seznámen s celým projektem. Pokud je to nezbytné, účastní se všech důležitých technických rozhodnutí. Druhý programátor je odpovědný za testování vyvíjeného produktu a provádí výzkum pro potřeby vedoucího programátora. Mohl by být nazván „viceprezidentem“ týmu, protože ačkoliv nemá rozhodovací pravomoc vedoucího programátora, musí mít stejnou úroveň znalostí a schopností a musí být v případě potřeby schopen vedoucího programátora plně nahradit.

Z hlediska požadované úrovně znalostí a schopností tvoří pozice vedoucího a druhého programátora vrchol. U žádné jiné funkce v týmu se současně nepožadují jak technické, tak manažerské schopnosti na tak vysoké úrovni. Kromě vynikajících technických schopností musí být například oba schopni vytvořit plán prací. Musí být schopni řídit, musí umět stanovit realistické cíle, rozdělit práci na úkoly a stanovit jejich priority. Úkoly musí přidělit lidem podle jejich schopností. Musí umět vyhodnocovat pokročilost prací na projektu a podávat zprávy nadřízeným organizačním složkám. Především vedoucí programátor musí mít navíc schopnosti komunikovat s lidmi, musí vědět, jak je motivovat k práci a musí se starat o udržování jejich vzájemných mezilidských vztahů.

Poznámka k překladu názvu funkce *backup programmer*: překlad „záložní programátor“ nevystihuje důležitost jeho postavení v týmu. Také překlad „zástupce vedoucího programátora“ vyvolává představu jeho silnější podřízenosti.

3. Knihovník (the programming secretary – PS, the librarian). Knihovník nebo také sekretář programátorského týmu je odpovědný za práci s dokumentací týkající se projektu. Udržuje její aktuální stav, zpřístupňuje informace těm, kteří je potřebují. Udržuje stav knihoven, testovacích dat a archivuje výsledky testů. Musí mít znalosti z oblasti zpracování textu (word processing) a psaní technických textů, musí ovládat metody a prostředky pro údržbu dokumentace. Měl by mít základní programátorské znalosti a měl by mít přehled o nástrojích používaných během prací na projektu.

4. Programátoři (the junior programmers – JP). Jednotliví programátoři realizují představy vedoucího programátora, plní jeho pokyny. Implementují části projektu, které jim byly přiděleny a jsou odpovědní za jejich funkčnost.

Ve srovnání s vedoucím a druhým programátorem nemusí mít řadoví (junior) programátoři univerzální schopnosti. Bývá pro ně typická spíše specializace na určitý druh problémů. Mohou být rovněž přechodně najímáni právě pro jejich speciální znalosti a dovednosti. Nejsou kladeny žádné požadavky na jejich manažerské schopnosti s výjimkou schopnosti vypracovat si časový plán vlastních prací a zorganizovat si vlastní práci tak, aby její výsledky zapadaly do časového plánu vedoucího programátora.

5. Správce projektu (the project administrator – PA). Součástí týmu může, ale nemusí, být správce projektu. Jeho úkolem je přebírat část administrativních povinností vedoucího programátora.

.REM [Mayr90–p.753,fig.15.7]

3.3 Komunikační struktura

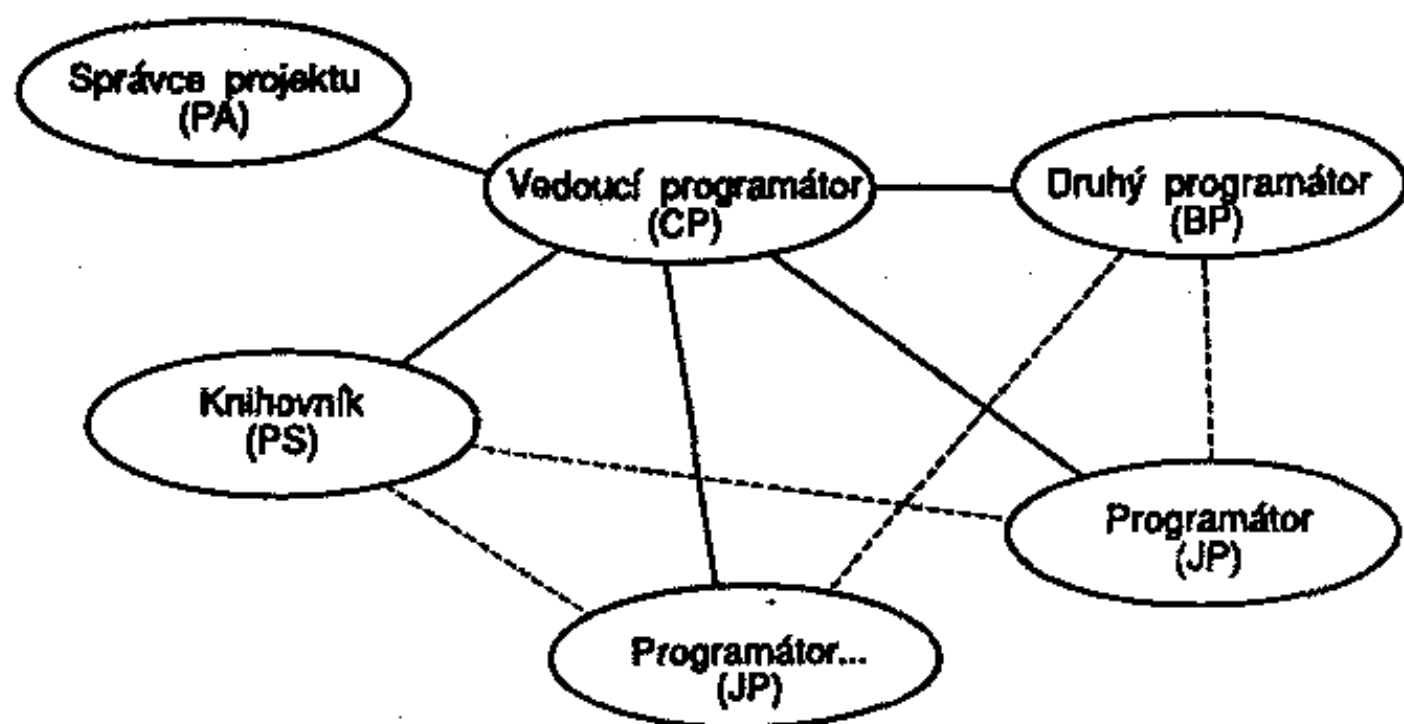
Komunikační struktura týmu má hvězdicovou podobu. Centrálnímu postavení vedoucího programátora při řízení a rozhodování odpovídá i jeho centrální postavení v rámci komunikační struktury (viz obr. 1).

Centrální postavení je sice výhodné z pohledu řízení, ale přináší i nevýhody v podobě vysokých komunikačních nároků na vedoucího programátora (přesycení komunikačních cest) a v podobě možného neuspokojení požadavků ostatních členů týmu (okrajové postavení v rámci komunikační struktury). Proto si i přes formálně výsadní postavení vedoucího programátora vytvářejí vedlejší komunikační vazby.

Vazby mezi programátory (JP) a druhým programátorem (BP) jsou vynuceny skutečností, že BP je odpovědný za testování. V období fází testování produktu se tyto vazby zesilují a BP se dostává do centrálního postavení. Vznik a oprávněnost těchto vazeb vyplývá i ze skutečnosti, že až na rozhodovací pravomoc je BP „dvojníkem“ vedoucího programátora.

K dalším pomocným vazbám dochází mezi programátory a knihovníkem (PS). Tyto vazby vyplývají z přístupu k dokumentaci, k testovacím datům a podobně.

Vytvoření pomocných komunikačních vazeb by však v žádném případě nemělo nepříznivě ovlivnit informovanost vedoucího programátora a nemělo by tak snižovat jeho rozhodovací schopnost.



Obr.1 Komunikační struktura týmu vedoucího programátora

.REM [Mayr90-p.755]

3.4 Potenciální problémy

Zvýraznění jedné oblasti nebo podcenění jiné může vést k problémům. Tato skutečnost je vyzvednuta v kritice koncepcie týmu vedoucího programátora (CPT) [Mccl81]. Jsou zde uvedena dvě základní nebezpečí.

Prvním z nich je očekávat od vedoucího programátora (CP) příliš mnoho, pokládat ho za jakéhosi „superprogramátora“. Úkoly spojené s projektem mohou v určitých situacích převyšovat schopnosti jednoho člověka. Problém přetížení lze zmírnit přepracováním plánu projektu, rozdělením části odpovědnosti CP mezi další členy týmu, ale vyhnout se mu lze jen omezením celkové velikosti projektu.

Druhé nebezpečí spočívá v centrálním postavení vedoucího programátora. Osobní vlastnosti CP mohou vést k autoritářskému chování, což přináší nebezpečí celkového snížení efektivity práce.

Styl řízení je nutné zvolit podle charakteru organizace (viz kapitola 2) a podle sehra- nosti týmu. Jeho charakter se může pohybovat od direktivního až po demokratický. Čím je tým sehra- nější, tím víc se lze přiblížit demokratickému stylu řízení bez nebezpečí snížení efektivity práce.

Další nebezpečí spočívá v postavení druhého programátora (BP). Ačkoliv se požadav- ky na jeho schopnosti neliší od požadavků na schopnosti CP, je mu odeprána rozhodovací pravomoc. Tato skutečnost u něho může vyvolávat (zvláště v kritických situacích) stres, což snižuje jeho schopnosti právě v době, kdy je to nejméně žádoucí.

U řadových programátorů (JP) může dojít ke ztrátě nadšení pro věc v okamžiku, kdy jim je přidělen úkol, na který nestačí. Pocity klivdy mohou vyplývat i z jejich neuspoko- jené touhy po profesionálním růstu. Začínající programátoři mohou být nadšeni tím, že dostávají přesně definované úkoly, ale po čase mohou začít pociťovat nedostatečné zapojení do procesu rozhodování o způsobech řešení projektu. Některým z nich může rovněž vadit snaha o redukci vzájemné komunikace s ostatními programátory na nezbytné minimum.

.REM [Mayr90-p.758]

3.5 Otázka profesionálního růstu

V oblasti profesionálního růstu existuje nebezpečí vyplývající přímo z typu organizace- týmu. Rozsah nebezpečí závisí na osobnostech jednotlivých členů. Pokud CP zjistí, že některý z řadových programátorů touží po zvyšování kvalifikace a po rozvoji vlastních schopností, může mu přidělovat odpovídající úkoly a zvyšovat podíl jeho odpovědnosti na projektu. Koncepce CPT neposkytuje formální návod, jak takové situace řešit. Přesto poskytuje prostředí CPT programátorům (JP) dobré možnosti pro zvyšování profesionál- ních kvalit. Jsou jim přidělovány úkoly různého typu a různé obtížnosti. JP se setkají se všemi typy činností, včetně činností CP a BP, administrativní činnosti nevyjímaje. Aby byla zajištěna efektivní komunikace mezi CP a JP, musí být i řadovým programátorům přidělovány úkoly (trénovací), jejichž úkolem je zvyšovat jejich odbornou úroveň až na úroveň znalostí CP či BP. Vedoucí programátor (CP) by tedy měl část času ostatních členů týmu vyhradit i na tyto neproduktivní činnosti. Součástí práce JP by měla být i účast na výuce. Odpovědnost za výuku ostatních může být delegována na vhodného člena týmu.

3.6 Shrnutí

Koncepce týmu vedoucího programátora se hodí nejlépe k řešení projektů omezeného rozsahu. Projekty mají mít hierarchickou strukturu, aby mohly být rozčleněny na pracovní jednotky podle schopností členů týmu a podle jejich zájmů. Do vedení týmu je nutné

dosadit zkušené lidi, jednoho s dobrými vůdčími schopnostmi, druhého se schopnostmi a se sklony být dobrým „viceprezidentem“. Lidi, kteří spolu dobře vycházejí, dobře spolupracují, kteří mají schopnosti komunikovat s ostatními, kteří mají schopnosti odhadnout motivační potřeby ostatních členů týmu a kteří jsou podle toho ochotni určovat náplň jejich práce. Je potřebný také jeden člen týmu se schopnostmi vyžadovanými pro práci knihovníka (programming secretary). Tato funkce nesmí být podceňována.

.REM [Mayr90-p.760]

4 Tým pracující s potlačením osobnosti (The Egoless Programming Team – EP)

4.1 Hlavní rysy

Koncepce tohoto týmu je úplným protikladem koncepce CPT. Je založena na volné spolupráci bez definice konkrétních rolí nebo komunikačních struktur. Každý je odpovědný za vše a každý člen týmu se snaží dosáhnout společného cíle. Úloha vedoucího týmu se přiděluje neformálním způsobem na základě mínění většiny členů týmu. V této funkci se střídají lidé podle momentálních potřeb týmu a podle řešeného projektu. Základní myšlenkou je poskytnout všem členům týmu co největší samostatnost. Tento typ týmu bývá někdy označován jako *demokratický tým*. Kombinuje v sobě myšlenku *programování s potlačením osobnosti* a demokratického pracovního prostředí. Programování s potlačením osobnosti (egoless programming) je založeno na zkušenosti, že je nebezpečné a neefektivní nechat programátory „sedět na svém programu“ a považovat tento program za část jejich já [Wein71]. Takový přístup ztěžuje odhalování chyb v algoritmech, protože jedinec má obvykle sklony k tomu, aby si vytyčil určitý směr a jen obližně mění úhel pohledu na své vlastní dílo. Program (nebo jeho část) je tedy považován za dílo kolektivu, je u něj potlačeno původní autorství. Tvůrce se vzdává dalších osobních nároků na své dílo a chová se k němu jako k majetku týmu.

Vzájemná výměna programů má příznivý vliv na jejich kvalitu a urychluje odhalování chyb. Další výhoda vyplývá ze skutečnosti, že odchod autora programu neohrozí práci na celém projektu. Přitom největším problémem v tomto případě nebývá zajistit zdrojové texty, ale myšlenkovou návaznost.

Výměnu kódu, jednu z hlavních zásad programování s potlačením osobnosti, lze praktikovat i u nedemokratických týmů. Ale výhody tohoto způsobu práce lépe vyniknou v demokratickém prostředí.

Za výhody tohoto typu týmu (je-li tým skutečně schráný) bývají považovány následující skutečnosti:

- zvyšuje se sebevědomí týmu; členové získávají pocit hrdosti na to, že jsou jeho součástí,

- práce jednotlivých členů je samostatnější než u CPT,
- práce členů týmu je pestřejší a zajímavější,
- práce v EP týmu přináší větší uspokojení,
- výsledné programové vybavení (software) je lepší.

.REM [Mayr90-p.761]

4.2 Požadavky na členy týmu

Protože žádný z členů týmu nemá výsadní postavení, musí všichni splňovat následující požadavky:

- musí se chovat demokraticky;
- výsledek hlasování musí být považován za závazný (ti, kteří hlasovali proti nesmí sabotovat práci ostatních);
- musí být schopni vyrovnávat se s neexistencí pevné struktury týmu;
- musí si osvojit schopnosti tvořit tým;
- musí být přístupní spolupráci a vzájemné komunikaci;
- musí být schopni pracovat na základě dohody a musí umět předcházet konfliktům.

4.3 Budování týmu

Při budování demokratického týmu lze i přes neexistující pevnou strukturu postupovat podle plánu. Členové týmu mají být seznámeni se záměry a s průběhem budování týmu. Demokratický proces však chce svůj čas a neobejde se bez konfliktů. Vývoj skupiny probíhá zhruba v následujících fázích (byly ponechány anglické názvy):

- **Formation** (formování). Vybírají se členové týmu.
- **Storming** (vybouření se). Členové týmu si hledají své pozice, prosazují své představy.
- **Norming** (normalizace poměrů). Během této fáze si skupina vytváří svá pravidla.
- **Performing** (výkonná fáze). Skupina spolupracuje, společným zájmem je dosažení vytyčeného cíle.

Přes demokratickou podobu týmu by už při jeho vytváření měly být stanoveny základy budoucích pravidel.

4.4 Potenciální problémy

První problémy vyplývají z nejednoznačnosti rolí v počátečních fázích tvorby týmu. V počátečních fázích může vznikat napětí u členů týmu, kteří jsou zvyklí pracovat víc než ostatní a pociťují, že nejsou odpovídajícím způsobem odměňováni.

Velmi dobrý kolektiv z hlediska soudržnosti nemusí mít automaticky lepší výkonnost.

Velkým problémem je jen velmi volné vedení; nikdo nemá rozhodovací pravomoc a nemůže proto ukončit často neplodné diskuse.

U tohoto týmu nemůže dojít k minimalizaci času potřebného na vzájemnou komunikaci, protože všichni řeší všechno. Skupina by proto měla být co nejmenší (2 až 7 lidí).

Během krizových situací, kdy je nutné určit vedoucí osobnost, je často těžké najít někoho, kdo by se tohoto úkolu ujal. Problém neexistujícího formálního vedoucího souvisí i s vnějšími kontakty skupiny (neexistuje reprezentant skupiny).

Struktura produktu odpovídá neformálním vztahům v rámci skupiny, závisí na vzájemných vztazích jednotlivých členů. Při vývoji projektu je nutné uspokojit mnoho různých názorů. Na druhé straně to vede k důkladnému promyšlení všech možností. Ukazuje se, že se i při značně rozdílných počátečních názorech nakonec dospěje k obecně lepšímu výsledku, než by tomu bylo u direktivně řízené struktury.

Členové týmu, kteří nemají sklony k samostatné práci, mohou ztrácet jistotu a motivaci k práci („nikdo jim neřká, co mají dělat“).

Mají-li někteří členové značné zkušenosti či znalosti, které se jim však nepodaří prosadit, mohou to pociťovat jako křivdu.

4.5 Profesionální růst

Myšlenka, že by takový typ týmu mohl být ideálním učebním prostředím, je pravdivá jen zčásti. Skupina často izoluje neproduktivní členy a přiděluje jim nevýznamné úkoly. Tímto stavem mohou trpět začínající programátoři.

Osvědčilo se stanovit si jako jeden z cílů nejen vývoj programového vybavení, ale také výukovou aktivitu. K takové činnosti však musí existovat určité pohnutky (motivace). Takovým motivačním nástrojem může být například finanční či prestižní cena za učitelský výkon.

.REM [Mayr90-p.764]

4.6 Shrnutí

Tým pracující s potlačením osobnosti má proti jiným týmovým koncepcím určité výhody ve smyslu velké samostatnosti členů. Vzhledem k slabému manažerskému řízení a vysokým požadavkům na vzájemnou spolupráci a komunikaci je tento typ týmu vhodný jen pro malé skupiny lidí, kteří mají dobré vzájemné vztahy. Tato skutečnost vede

k nutnému omezení velikosti projektu. Katastrofální následky může mít volba tohoto typu týmu v případě kolektivu složeného z lidí, kteří si nejsou schopni stanovit realistické cíle a potřebují vedení. Zdá se, že tato koncepce je nejméně úspěšná za podmínek:

- je-li skupina malá (2 až 7; nemusí spolu komunikovat větší množství lidí);
- považují-li se členové týmu za vzájemně rovnocenné;
- mají-li členové týmu dostatečné zkušenosti; jsou-li schopni vytyčit si své cíle a dosáhnout jich.

5 Chirurgický tým (The Surgical Team – ST)

5.1 Hlavní rysy

Struktura chirurgického týmu je podobná struktuře CPT. Je rovněž založena na specializaci, ale přistupuje k ní jiným způsobem.

5.2 Složení týmu

Členové týmu mají přiděleny následující role (pro jednoznačnost jsou uvedeny anglické názvy):

1. **Surgeon (SG – chirurg)**. Chirurg je technickým vedoucím týmu. Ve srovnání s vedoucím programátorem (koncepce CPT) přísluší chirurgovi méně administrativních povinností. Tyto povinnosti jsou přeneseny na administrátora. Tím je dosaženo snížení celkových požadavků na chirurgovy schopnosti.

2. **Copilot (CP – asistent chirurga)**. Povinnosti asistenta chirurga odpovídají povinnostem druhého programátora (BP) v CPT. Asistent chirurga je navíc odpovědný za udržování kontaktů s ostatními týmy. Tato funkce navíc opět odlehčuje zátěž kladenou na chirurga.

3. **Administrator (A – administrátor, jednatel, tajemník)**. Administrátor je odpovědný za personální otázky členů týmu, za rozpočet, za provozní prostředí (místnosti, počítačový čas a podobně) a za styk s nadřazenými složkami. Jeho role nevyžaduje příliš velké technické schopnosti, ale požadují se dobré znalosti a zkušenosti administrativní, schopnosti plánovací a schopnosti vést jednání.

4. **Editor (E – dokumentarista, správce archívu)**. Editor vytváří veškerou projekto-
vou dokumentaci, z čehož vyplývá požadavek na schopnost psát technicky orientované texty. Musí být schopný komunikovat s ostatními členy týmu, měl by s nimi udržovat dobré vztahy. Vyžaduje se od něj formální důslednost a pečlivost.

5. **Secretary (SE – sekretář/ka)**. Požadují se znalosti psaní na stroji. Jeden nebo dva členové týmu s tímto pracovním zařazením mají za úkol pomáhat administrátorovi a editorovi.

6. **Programming clerk (PC – knihovník)**. Plní funkci odpovídající funkci knihovníka v CPT (PS) s výjimkou činností převedených na editora.

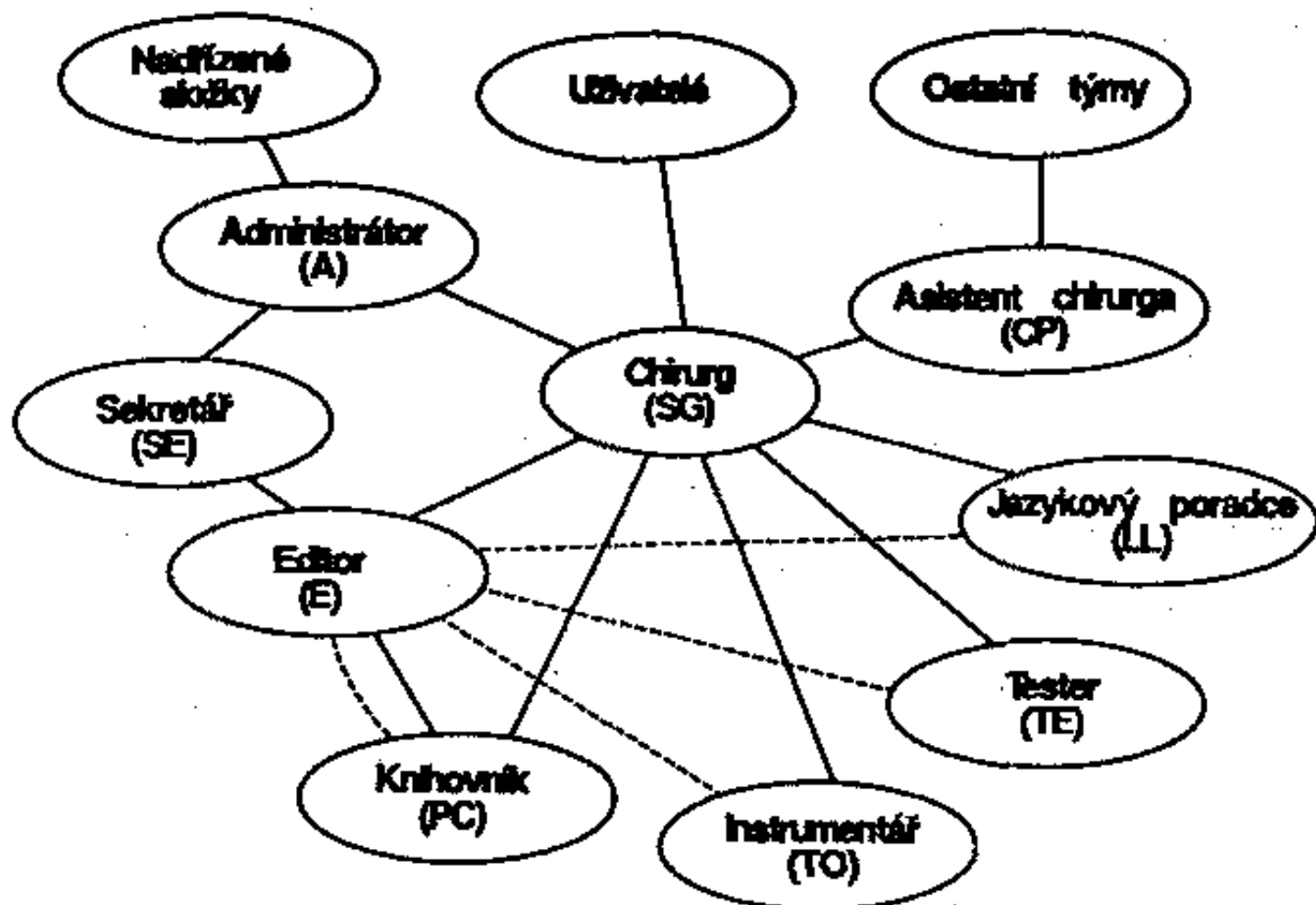
7. **Toolsmith (TO – instrumentář, nástrojař)**. Tento člen se specializuje na údržbu nezbytných technických (programátorských) nástrojů a prostředků (například ladicí prostředky) a provádí údržbu knihoven. V koncepci CPT nebyl pro tyto činnosti zvlášť vyhrazen člen týmu. Jeho činnost vykonával částečně knihovník (PS) a část úkolů byla přidělována řadovým programátorům (junior programmers). Instrumentář musí být dobrým systémovým programátorem a musí mít schopnosti komunikovat s lidmi. Spolupracuje s editorem při vytváření dokumentace v tom smyslu, že mu dodává požadavky a podklady na dokumentaci, která má být vypracována.

8. **Tester (TE)**. Tester je osoba odpovědná za realizaci testovacího plánu, který vypracoval chirurg. Provádí také vyhodnocování testů a poskytuje jejich výsledky ostatním členům týmu. Nemusí mít schopnosti aktivně hledat a vytvářet algoritmy, ale požadují se od něho dobré analytické schopnosti. Musí být trpělivý, pečlivý a důsledný (puntičkář). Musí mít talent rozbít program na menší části a hledat kritické případy, kdy by mohlo docházet k chybám. Měl by mít sklony stát v opozici k tvůrcům programu a chovat se k produktu co nejméně očekávaným způsobem.

9. **Language lawyer (LL – jazykový poradce)**. Tento člen týmu je specialistou na programovací jazyk, používaný týmem při řešení projektu. Dokonale zná detaily programovacího jazyka a dokáže využít jeho předností k co nejefektivnější implementaci řešených problémů.

5.3 Komunikační struktura

Komunikační struktura v zásadě vychází z koncepce CPT (viz obr.2). Základní struktura je hvězdicová, ale objevují se i další komunikační vazby. V centru zůstává chirurg, jako technický vedoucí projektu. Z hlediska komunikace je další významnou osobou editor. Konfliktní postavení v této struktuře má člen v roli sekretáře/ky. Není podřízen chirurgovi a v případě jeho většího zatížení (a je-li v týmu jen jeden sekretář) není jasné, zda má dávat přednost plnění požadavků ze strany editora či administrátora.



Obr.2 Komunikační struktura chirurgického týmu

5.4 Výhody a potenciální problémy

V porovnání s týmem vedoucího programátora (CPT) jsou role členů chirurgického týmu přesněji definovány. Komunikace s okolím týmu je rozdělena mezi více členů (viz obr.2). Za styk s nadřizenými složkami je odpovědný administrátor, za kontakt s ostatními týmy odpovídá asistent chirurga a stykem s uživateli je pověřen chirurg. Vnitřní komunikace leží z větší části na editorovi. Kladem tohoto uspořádání je, že za každou oblast komunikace odpovídá jediná osoba. Tím se snižuje riziko nedorozumění a nejednoznačnosti.

Potenciální problémy vyplývají z postavení administrátora. Při jednání s nadřizenými složkami mu může chybět chirurgova rozhodovací pravomoc.

Další problémy vyplývají z úzké specializace. Ta může postupně vést k pocitu stereotypu a může docházet ke ztrátě motivace k práci. Na druhé straně vede specializace ke zvýšení rychlosti řešení projektu a k jeho vyšší kvalitě. Tento druh specializace také vede ke zvýšení samostatnosti práce členů a ke snížení vzájemných komunikačních nároků.

Členové týmu se specializují jen na svou oblast a chybí jim přímý kontakt s ostatními druhy činností. To může vést k neuspokojení touhy po získávání širšího rozhledu, který je pro profesionální růst nezbytný.

Tato koncepce nedefinuje činnosti, zabývající se učením a tréninkem. V koncepci chybí práce se začínajícími, řadovými (junior) programátory.

5.5 Shrnutí

Chirurgický tým je velmi různorodá skupina. Specializace zde dosahuje velmi vysokého stupně. Výsledkem této skutečnosti je značná koncentrace znalostí, což se odráží ve vysoké kvalitě výsledného produktu. Skupina však není příliš pružná – jednotliví členové jsou obtížně nahraditelní.

6 Revidovaný tým vedoucího programátora (The Revised Chief Programmer Team – RCPT)

6.1 Důvody přehodnocení CPT

Tato týmová koncepce [McCl81] představuje další vývojový stupeň týmu vedoucího programátora. Snaží se odstranit jeho následující nedostatky:

- přetížení vedoucího programátora,
- neodpovídající stupeň samostatnosti druhého programátora,
- projekt je příliš závislý na kvalitách některých členů týmu (především na kvalitách vedoucího programátora),
- odpovědnosti jednotlivých členů nejsou dostatečně definovány,
- vedoucí programátor má příliš velkou moc.

6.2 Složení týmu

Složení týmu z větší části odpovídá struktuře týmu vedoucího programátora. McClure navrhuje přejmenovat role vedoucího a druhého programátora na vedoucího projektu (project leader) a zástupce vedoucího (coleader). Paní McClure (autorka koncepce) navrhuje zavedení dalších dvou funkcí:

- **administrativní pracovník (administrator)** – přebírá veškerou administrativní činnost, která příslušela vedoucímu programátorovi.
- **pracovník pro styk s uživateli (user liaison)** – odpovídá za styk se zákazníky, je reprezentantem týmu vůči okolnímu prostředí. Součástí jeho povinností je také návrh testovacího plánu. Vedoucí projektu tento plán pouze kontroluje.

Zavedení těchto funkcí umožňuje odlehčit obě technické vedoucí osobnosti, umožňuje přesněji definovat komunikační vazby a odpovědnosti. Uvedení čtyři členové (project leader, colleader, administrator, user liaison) tvoří jádro týmu.

6.3 Výhody koncepce RCPT

Struktura přestává být striktně hierarchická a umožňuje snížit požadavky na schopnosti vedoucí osobnosti (podobný krok je patrný i u chirurgického týmu). Funkce v jádru týmu jsou rozděleny rovnoměrněji, což vedoucímu projektu ztěžuje případné zneužití postavení. Vedoucí projektu a jeho zástupce se mohou více věnovat technické stránce projektu. Pro role administrativního pracovníka a pracovníka pro styk s uživateli lze najít lidi, kterým tato činnost vyhovuje lépe než, obvykle více technicky založenému, vedoucímu projektu nebo jeho zástupci. Ve srovnání s chirurgickým týmem vede menší specializace programátorů (JP) k větší pružnosti týmu a k větší zastupitelnosti jednotlivých členů. Přestože ani RCPT nedefinuje proces zdokonalování (výuky) jednotlivých členů, jsou pro tuto činnost v tomto typu týmu podmínky lepší než v chirurgickém týmu.

6.4 Shrnutí

Stejně jako v předchozích typech týmu se i u tohoto typu doporučuje zavést vhodný druh vzdělávání. Ze všech zmíněných struktur týmu se koncepce RCPT zdá být nejvyváženější. Je-li skupina složena z lidí s vyváženými zájmy a schopnostmi, pak se tato koncepce jeví jako nejvhodnější.

7 Závěr

V čem se tedy teorie a praxe shodují? V současné době je obecně přijímána skutečnost, že by softwarový projekt neměl být řešen jedním velkým týmem. Naopak, programátorský tým má být relativně malý. Teorie tvrdí, že by počet členů neměl být větší než 20. Většina zkušených manažerů říká, že by tento počet neměl být větší než 8 [Mayr90], [Somm89], [Lamm89].

Každé teorii je vlastní snaha o zjednodušení a zobecnění pohledu. Teoretické závěry tedy nemusí plně vystihovat všechny konkrétní praktické případy.

Pokud se tento příspěvek zmiňuje o několika odlišných koncepcích programátorského týmu, znamená to jen tolik, že se při zobecňování praktických případů dospělo k těmto koncepcím jako k určitým reprezentantům. Jednotlivé koncepce popisují stav, ke kterému pravděpodobně dojde jen výjimečně. Praxe by však měla být opatrná při odsuzování teorií, které jí mohou přinést užitek.

Většina programátorských týmů si pravděpodobně sama najde svou cestu. Teorie má být jen vodítkem, ale praxe by si neměla dovolit její závěry ignorovat nebo dokonce postupovat proti nim.

Literatura

- [Mayr90] Mayrhauser, A., *Software Engineering: Methods and Management*, Academic Press, San Diego, CA, 1990, str.731–795
- [Somm89] Sommerville, I., *Software Engineering*, Addison–Wesley Publishing Company, 1989, str. 477–492
- [Lamm89] Lammers, S., *Programmers at Work*, Tempus Books of Microsoft Press, Redmond, WA, 1989
- [Mill72] Mills, H., Baker, F., „Chief Programmer Teams“, *IBM Systems Journal* 11, 1/1992, str.56–73
- [Mill83] Mills, H., *Software Productivity*, Little, Brown & Co., 1983
- [Broo75] Brooks, F., *The Mythical Man Month*, Addison–Wesley, Reading, MA, 1975
- [Mccl81] McClure, C., *Management of Software Development and Maintenance*, Van Nostrand Reinhold, New York, NY, 1981
- [Wein71] Weinberg, G. M., *The Psychology of Computer Programming*, Van Nostrand Reinhold, 1971
- [Knud79] Knudson, H., Woodworth, R., Bell, C., *Management—An Experiential Approach*, McGraw–Hill Series in Management, 1979

Author: Ing. Petr Pfikryl
Vysoké učení technické Brno
Fakulta elektrotechnická
Katedra informatiky a výpočetní techniky
Božetěchova 2, 612 66 Brno
Tel. : (42) (5) 746 111/231
Fax : (42) (5) 750 252
e-mail: prikryl@dcsc.fec.vutbr.cs