

# Vývojové tímy v softvérovom inžinierstve

JÁN ŠNIRC

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

**Abstrakt.** Vývoj softvéru je zložitý proces, ktorého úspech závisí od viacerých aspektov. V technickej rovine sú to použité technológie, nástroje a metodológia, na druhej strane spôsob organizácie tímu a komunikácie medzi jednotlivými členmi. Cieľom tejto eseje je prezentovať základné typy organizácie tímov, ich výhody, nevýhody a vhodnosť pre jednotlivé metodológie. Na záver uvádzam vlastné skúsenosti z pôsobenia v jednom tíme.

## Úvod

Spôsob organizácie vývojového tímu v softvérových projektoch závisí od viacerých faktorov. Od komplexnosti projektu, ľudí, veľkosti tímu, dĺžky jeho existencie až po použitú metodológiu pri vývoji softvéru. Práve použitá metodológia výrazným spôsobom určuje štruktúru tímu. Avšak treba povedať, že jej výber ovplyvňujú tiež predošlé faktory, takže ako to už býva, všetko so všetkým súvisí.

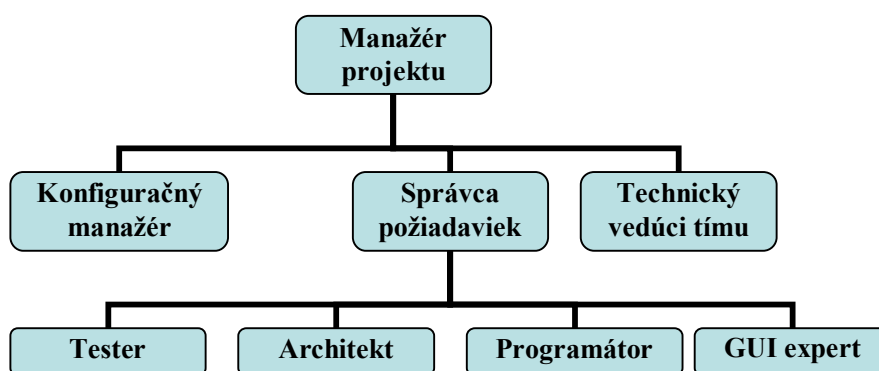
V hrubom priblížení rozoznávame tri základné typy vývojových tímov [2]: sekvenčný, skupinový a sieťový typ. Každý z nich reprezentuje odlišný pohľad na organizáciu tímu, komunikáciu medzi jednotlivými členmi ako aj samotný vývoj softvéru. Jednotlivé typy podrobnejšie opíšem v nasledujúcom texte.

## Sekvenčný typ

Dobrá proces vedie k dobrému výsledku. To je základná myšlienka, ktorá ovplyvňuje formovanie vývojárskeho tímu do sekvenčného typu. Vývoj softvéru vychádzajúci z tohto presvedčenia je prísne lineárny a riadený úlohami, čo sa odráža aj na štruktúre tímu (Obr. 1). Tá je formalizovaná a založená na rolách. Jednotliví pracovníci sú špecialistami na konkrétne oblasti, v ktorých samostatne riešia pridelené úlohy a sú ohodnotení presne za to, čo vyprodukujú. Keďže vývoj je riadený úlohami, ktoré sú dopredu dané, potreba diskusie medzi ostatnými členmi tímu je minimalizovaná

*Manažment v softvérovom inžinierstve, máj 2005, s. 1-7.*

a v prípade potreby prebieha iba cez formálne kanály vertikálnym smerom. To implikuje slabé sociálne väzby medzi členmi tímu a ich jednoduchú výmenu. Ďalšou výhodou takejto organizácie, ktorá sa začne prejavovať pri väčšom počte ľudí je jednoduché rozdelenie tímu na oddelenia a riadenie, čím je nedochádza k nedorozumeniam pri riadení a komunikácii. Z pohľadu členov tímu je takáto štruktúra vhodná ak sú všetci skúsenými odborníkmi na pridelenú oblasť. V prípade menej skúsených ľudí, ktorý potrebujú doplniť svoje znalosti a jednotlivé kroky konzultovať nastáva značná neefektivita, pretože komunikácia v sekvenčnom type je už z jeho podstaty obmedzená a pomalá.



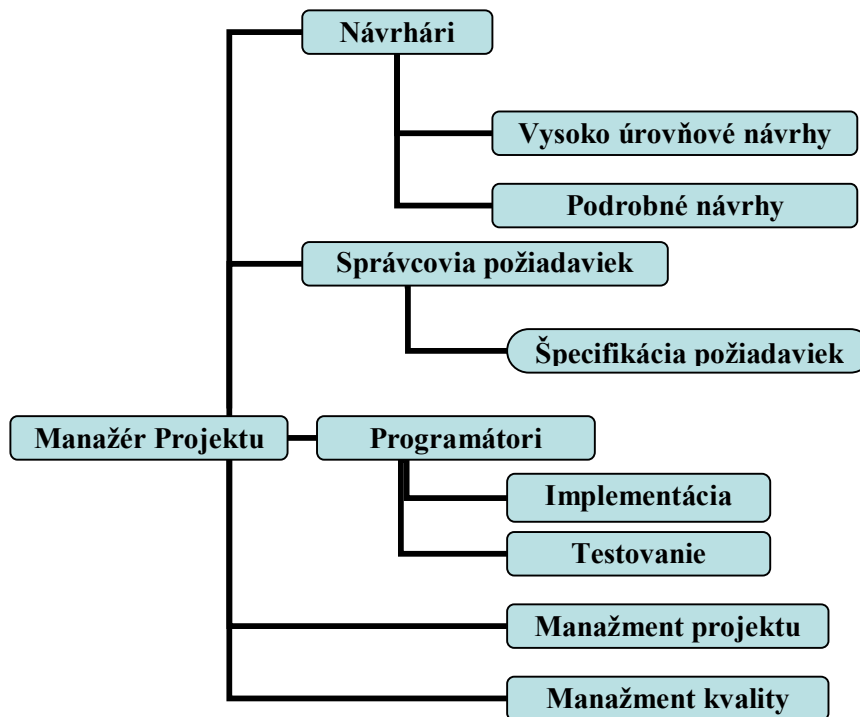
Obr. 1 Centrálne riadený tím

Typickým prípadom kedy je táto forma organizácie použitá je vodopádový model vývoja softvéru. Jednotlivé fázy vodopádového modelu sa vykonávajú v striktnom poradí, bez prekrývania či iteratívnych krokov.

## Skupinový typ

Vývoj softvéru v skupine je založený na úlohách, avšak na rozdiel od sekvenčného typu ich riešia všetci členovia skupiny a to iteratívne. Už samotný fakt, že vývojári pracujú v skupine na spoločnej úlohe implikuje silné sociálne väzby, ktoré okrem formálnej komunikácie ústia aj do neformálnej. Sila skupiny spočíva práve v komunikácii medzi členmi a vzájomnom zdieľaní svojich znalostí. Vývoj v skupine však môže znamenať aj vznik konfliktov medzi členmi a zníženie produktivity. Z tohto dôvodu je potrebné zaviesť pravidlá, ktoré predchádzajú vznikom konfliktov a venovať dodatočné úsilie na vytvorenie priateľskej atmosfére v skupine

Príkladom skupinového typu sú špirálový a evolučný prístup k vývoju softvéru. Ten je rozdelený na iterácie, ktorých výstupom je pridaná nová funkcionality. Výsledkom iterácie sú tiež získané poznatky, ktoré sú použité v nasledujúcej iterácii.



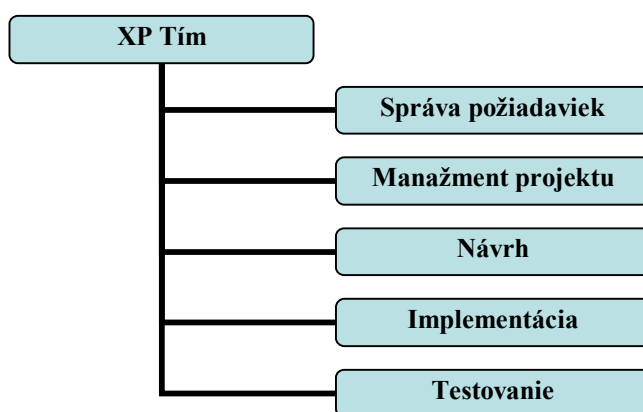
Obr. 2 Decentralizovane riadený tím

Vývoj softvéru je riadený decentralizovane (Obr. 2). Na čele tímu je definovaný koordinátor, ktorý riadi niektoré úlohy a lídrov zodpovedných za príslušné úlohy. Komunikácia medzi skupinami a jednotlivcami prebieha horizontálne, medzi koordinátorom a lídrami vertikálne.

Relatívne nový prístup k vývoju softvéru predstavujú agilné metódy. Základné myšlienky sú vyjadrené v manifeste agilného vývoja softvéru [3]. Z pohľadu organizácie softvérových tímov je dôležitým faktom, že agilné metódy upriamujú pozornosť na jednotlivcov a ich vzájomnú komunikáciu. Preferované sú blízke vzťahy v tíme a uzavreté pracovné prostredie a iné aktivity, ktoré zvyšujú morálku v tíme.

Jedným z derivátom agilných metód je extrémne programovanie, ktorého autorom je Kent Beck. Ide o zjednodušenú metódu vývoja softvéru založenú princípoch jednoduchosti, komunikácie a spätnej väzby. K zaujímavým prvkom tejto metódy patrí myšlienka zdieľaného kódu. Každý člen vlastní celý kód tímu, rozumie mu a môže ho

flexibilne meniť. Na to, aby princíp zdieľaného kódu fungoval je potrebné samozrejme zaviesť konvencie písania kódu. Ďalším prvkom, ktorý ovplyvňuje vnútornú štruktúru skupiny je zavedenie dvojíc programátorov, ktorí spolu úzko spolupracujú pri implementácii a vzájomne sa dopĺňajú. XP tím je organizovaný demokraticky [Obr. 3], t.j. neexistuje žiadny stály líder a komunikácia medzi členmi tímu prebieha horizontálne. Výhodami takejto organizácie je prispievanie každého člena k rozhodnutiam, učenie sa jeden od druhého, čím sa zvyšuje spokojnosť s prácou. Okrem štandardných nevýhod, ktoré so sebou prináša práca v skupine sa pridáva aj zníženie osobnej zodpovednosti v dôsledku demokratického rozhodovania.

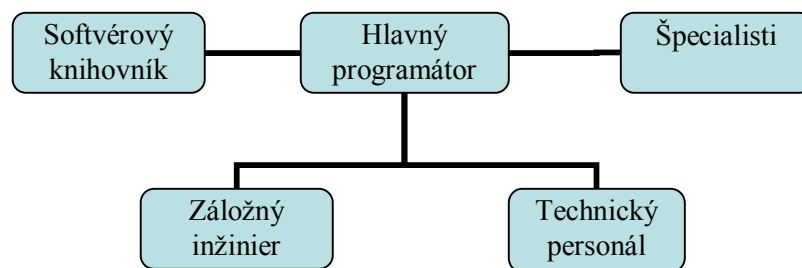


Obr. 3 Organizácia XP tímu

### Sieťový typ

V centre pozornosti sa nenachádza proces, ale produkt. Sieťový typ softvérového tímu vychádza z presvedčenia, že skúsení ľudia tvoria dobrý softvér. Tím tvorí sieť s uzlami, ktorými sú skúsení programátori a koncovými bodmi, ktoré tvoria menej zdatní členovia tímu, pričom vytvorená sieť nemusí mapovať organizáciu spoločnosti, či geografickú polohu členov. Úlohy, ktoré treba riešiť nie sú dopredu dané, ale vznikajú z komunikácie v sieti, ale existujú aj centrálné úlohy ako je napr. správa verzií, tvorba dokumentácie. Výhodou tohto typu je rýchly vývoj aplikácii v tíme kde sú skúsenosti jednotlivých členov na rôznej úrovni a fluktuácia menej skúsených členov je relatívne vysoká. Na druhej strane odchod človeka, ktorý tvorí uzol siete znamená pre projekt významnú stratu, nielen pre jeho odborné znalosti, ale aj pre väzby s ostatnými členmi tímu, ktoré bude musieť jeho nástupca nanovo nadväzovať. Ďalšou nevýhodou je komunikácia na diaľku, ktorú treba dodatočne podporiť nástrojmi a môže byť zdrojom nedorozumení.

Typickým príkladom môže byť organizácia tímu vyvíjajúca open source softvér, alebo tím hlavného programátora [1] (Chief programmer team). Ten vznikol v prostredí IBM v 70. rokoch ako odpoveď na existenciu značných rozdielov medzi vývojármi a vysokou fluktuáciou tých menej skúsených. Tím je postavený okolo skúseného hlavného programátora a ostatný členovia tímu tvoria pre neho technickú a netechnickú podporu (Obr. 4).



**Obr. 4 Organizácia tímu hlavného programátora**

Riadenie tímu je prísne centralizované. Riešenie hlavných problémov a koordinácia ostatných členov je v kompetencii hlavného programátora, ktorý tvorí jadro tímu. Ako podporu má pridelených dvoch až piatich pracovníkov na jeho podporu. Jedná sa o ostatných programátorov, dokumentaristov a softvérového knihovníka, ktorý môže pracovať pre viaceré tímy a stará sa o softvérovú konfiguráciu, asistuje pri výskume a príprave dokumentácie.

Výhodou je centralizované rozhodovanie a redukcia zbytočnej komunikácie. Na druhej strane je celý tím závislý od hlavného programátora a jeho správania sa, čo znižuje morálku v tíme.

### **Všeobecné princípy organizovania tímu**

Bez ohľadu na typ organizácie tímu existujú všeobecné princípy, ktoré môžu zvýšiť efektivitu tímu.

- Menej ľudí, ale kvalitných
- Plniť úlohy vzhľadom na možnosti a motiváciu v tíme

- Vyberať ľudí tak aby výsledkom bol harmonický tím
- Ak niekto nezapadne do tímu, radšej ho vymeniť
- Brooksov zákon – pridanie nových ľudí do projektu, ktorý mešká spôsobí jeho ďalšie oneskorenie.

### **Osobné skúsenosti z práce v tíme**

V súčasnosti som členom tímu, ktorý vyvíja softvér a preto som sa rozhodol prezentovať spôsob jeho organizácie a výhody resp. nevýhody z nej prameniace.

Projekt predstavuje zadanie riešené v rámci predmetu Tímový projekt inžinierskeho štúdia FIIT STU. Samotné zadanie projektu ako aj fakt že medzi členmi tímu existovali už predtým neformálne vzťahy vyústil do organizácie tímu, ktorá najlepšie zodpovedá typu skupina. Počas riešenia projektu používame princípy definované v agilných metódach a to najmä úzku spoluprácu medzi členmi tímu, tvorbu softvéru bez dokumentácie a pripravenosť na zmeny. Riadenia tímu je demokratické bez stáleho tím lídra. Samotný projekt bolo možné intuitívne rozdeliť na viaceré pod úlohy a k nim prislúchajúceho lídra, ktorý viedol diskusiu ale rozhodnutie bolo nakoniec tímové a súhlasili s ním všetci členovia.

Výhodou takejto organizácie je jednoznačne zdieľanie informácií v tíme a pri výskyte nových problémov brainstorming, ktorý generuje množstvo zaujímavých spôsobov riešenia. Problémom, ako som sa sám presvedčil môže byť demokratické rozhodovanie o riešení problému. V našom prípade išlo konkrétne o návrh komunikačného rozhrania. Predstavy členov tímu boli značne odlišné a všetky navrhované riešenia v princípe správne. To spôsobilo približne dvojtýždňové meškanie tejto úlohy a pri pohľade späť by v tomto prípade bolo autoritatívne riadenie vhodnejšie.

### **Záver**

Určenie najlepšieho spôsobu organizácie vývojového tímu, ako vyplýva z prechádzajúceho textu nie je triviálna záležitosť. Voľba resp. kombinácia uvedených typov záleží od riešeného projektu, znalostí členov tímu ich schopností vzájomne komunikovať, osobnej skúsenosti z práce v tíme atď. Cieľom príspevku bolo preto najmä priblíženie faktorov, ktoré tento výber ovplyvňujú.

### **Použitá literatúra**

1. Baker, F. Chief programmer team management of production programming. IBM Systems Journal 11, 1 (Jan. 1972), 56–73.

2. Sawyer, S. : Software development teams. Communications of the ACM December 2004 Vol. 47, No. 12
3. Manifesto for Agile Software Development, <http://agilemanifesto.org/> Marec 2005

### **Annotation**

#### *Software development teams*

Software development is a complex process and its result depends on many aspects. In technical level that are technologies, tools and methodology. On other side the way of organizing development team and communication among team member. The goal of this work is to present basic archetypes of development team structure, their advantages, disadvantages and suitability for different methodologies. At the end I introduce my own experiences from working in two different team types.