

Изабрани проблеми управљања тимом

Развој софтвера (9)

проф. др Милан Гњатовић

Факултет за компјутерске науке
Универзитет Џон Хезбит

17.05.2016.

Садржај

- 1 Увод
- 2 Проблематични програмер
- 3 Завршне напомене

Зашто је тимски рад важан?

- Развој софтвера се суштински своди на (квалитетни) тимски рад (без обзира које моделе и процесе развоја користите).
- Тимски рад подразумева:
 - укључивање колега у рад,
 - конверзацију,
 - толерисање разлика,
 - решавање проблема,
 - ...

Шта карактерише „проблематичног“ програмера?

- Непрофесионални (лични) став према послу.
 - Одбијање да прати план и смернице дефинисане дизајном,
 - одбијање да адекватно тестира софтвер,
 - ...
- Мала продуктивност узрокована нестручношћу.
 - Чак и између искусних програмера постоје велике разлике у продуктивности.
 - Сматра се да је однос времена потребних да различити програмери реше исти проблем бар 10:1.
 - Поред тога, постоји и група програмера која не може да реши проблем уопште.
- „Негативна“ продуктивност.
 - Неспособност програмера да прати план и смернице дизајна, отклони грешке у коду, и поуздано процени учинак, узрокује додатни посао за друге чланове тима.

Процена

- Сматра се да 10% професионалних програмера спада у категорију негативне продуктивности.
- Шта то значи?
 - Нпр., у тиму који се састоји од 7 случајно изабраних програмера, вероватноћа да ће бар један од њих бити проблематичан је већа од 50%.

Како препознати проблематичног програмера?

- Радије сакрива сопствену нестручност, него учи од других чланова тима (нпр., одбија да правовремено стави кôд на увид колегама).
- Одбија да препусти другим члановима тима да исправе грешке у његовом кôду (чак и кад то успорава или зауставља рад остатка тима).
- Жали се на одлуке које је тим знатно раније и коначно донео, и често почиње дискусију у вези са тим.

Зашто га треба уклонити из тима?

- Проблем лежи или у личном ставу програмера или у његовој нестручности. У оба случаја, узрок није лако уклоњив (а понекад га није ни могуће уклонити).
- Што дуже овакав програмер остаје у тиму, то:
 - добија више легитимитета у очима других чланова тима,
 - постаје одговоран за све веће делове кода,
 - све више утиче на рад осталих чланова тима.
 - Због тога, све га је теже уклонити из тима.

Превенција

- Рана контрола дизајна и кода током које се могу идентификовати некооперативни чланови тима, тј. они који:
 - не желе да деле свој рад,
 - не прихватају савете колега,
 - не желе да учествују у прегледању рада својих колега, итд.
- Контрола дизајна и кода пружа и секундарну корист:
 - Проблем који се типично јавља код проблематичних програмера је да једино они разумеју свој код.
 - Контролом се постиже да су са сваким делом софтвера упозната бар два члана тима. Тиме се избегава да развој зависи искључиво од једног члана тима.
 - Зато се одбијање програмера да учествује у контроли сматра неприхватљивим ризиком.

Шта треба да имате на уму?

- Аспект руковођења тимом који највише узнемирава чланове тима су вође који нису спремне да директно и ефективно обрачунају са члановима тима који нису кооперативни или раде само за себе.
- Вођа тима по правилу мисли да његов тим функционише боље него што то мисле чланови тима.

Литература

- McConnell, S., Problem programmers, *IEEE Software* (March/April 1998), pp. 126–128., 1998.
- Ferdinandi, P.L. Facilitating Communication. *IEEE Software* (September/October 1998), pp. 92–96, 1998.

Хвала на пажњи

- Питања су добродошла.