

Princípy tvorby softvéru

Project management, tvorba odhadov

Robert Lukočka

lukotka@dcs.fmph.uniba.sk

www.dcs.fmph.uniba.sk/~lukotka

M-255

- Zahájenie projektu
- Príprava kontraktu
- Manažment všetkých aspektov projektu:
Manažérsky cyklus: plánovanie, organizácia, vykonávanie, monitorovanie, vyhodnotenie
- Rozhodovanie
- Management informácií, riešenie konfliktov, ...

- definovanie plánovaných položiek
- estimácia (náročnosť, predpoklady, ...)
- alokácia zdrojov
- scheduling
 - critical path analysis
- update plánu

Treba plánovať rôzne

- rozsahy
project plan, release cycle plan, iteration plan
- aktivity
všetky v rámci SE; vyberám Contract management; Risk Management; Change management;

Zorganizovať

- Project team structure
- Definovať procesy
- Prostredie
- Zabezpečiť zdroje
- Priradiť zdroje abstraktným úlohám

- Tradičná štruktúra - tímy podľa technológií
- DevOps team structure - tímy podľa produktov / serviceov.

- + Aplikácia má tendenciu kopírovať štruktúru tímu
- + Zdieľaná zodpovednosť za produkt
- + Jednoduchšia komunikácia
 - Zastupiteľnosť
 - Špecializácia

Regulovať a riadiť

- procesy
- zdroje

Načo slúžia definované procesy?

- Meranie
- Uchovávať logy z merania (napr. testing scores, fixing scores)
- Výsledky použiť pre
d'alšie plánovanie, zlepšovanie procesov, ohodnotenie, ...

Spriaznené disciplíny:

- Interpersonal communication, conflict management and resolution
- Contract Management
- Risk Management
- Change Management
- Requirements management
- ...

Issue:

- bug
- požadovaný feature
- úloha
- chýbajúca dokumentácia
- ...

Pôvod v bug tracking

Issue:

- Identifikátor
- Typ (bug, feature)
- Status (open, in progress, resolved, closed, reopened)
- Priorita
- Komu je priradená
- Ďalšie potrebné informácie (popis bugu, referencia na požiadavky, číslo commitu)
- Komunikácia

Je dobré ak je issue tracking prepojený s version control.

- Redmine issues of redmine
- Ukázková issue

- Sprint Backlog
- Príklad
- Kanban boards
 - Príklad

Správne estimácie sú potrebné pre:

- plánovanie (čas, zdroje)
- rozhodovanie
- monitorovanie
- evaluáciu (estimácia vs. skutočnosť, napr. kontrakt)

Estimácie sú bežne zle interpretované:

- Estimácia nie je deadline - ak je estimácia iba jedno číslo, zmysel dáva stredná hodnota.
- Tlak od: Management, customers, executives (MCE)
- Potreba vytvorenia plánu projektu (zmluva / rozhodovanie zákazníka)
- Risk management (osobné ponímanie) - transfer risk to somebody else

Na druhej strane, estimácie udávané podliehajú rôznym psychologickým biasom a sú vychýlené.

- Nie je možné poskytnúť presné odhady v prvých fázach projektu (pri štandardnom prístupe sa vtedy robia zmluvy a určuje cena)
- Dobrým managementom projektu je možné urýchliť pokles neistoty
- Po odstránení najväčších neistôt by ideálne v prospech odhadov mal fungovať law of big number (ak zvládneme poskytovať nevychýlené estimácie).

Cone of uncertainty

I can see how I'd do it if I were rewriting that whole controller from scratch, but that would take days ... is there an elegant hack where I can change the inputs to this function in such a way that I don't have to rewrite its code? ... what if I monkeypatch it at the class level? ... wait, maybe there's an API call that almost does what I want, then I can tweak the results - hang on, what if I outsource it via an asynchronous call to the external OS? In that case I can confidently estimate that this will require less than two hours of typing. However, working out what to type is going to take me/us anywhere from one hour to several days. Sorry.

Ak je estimácia iba jedno číslo, zmysel dáva stredná hodnota.
Dobrá estimácia je nevychýlená.

- Hofstadter's Law:

It always takes longer than you expect, even when you take into account Hofstadter's Law.

- 90-90 Rule:

The first 90 percent of the code accounts for the first 90 percent of the development time. The remaining 10 percent of the code accounts for the other 90 percent of the development time.

Software estimates are always wrong, because the tasks being estimated are always, to some extent, terra incognita, new and unknown. However, sometimes the errors are in your favor; an obscure API, a third-party library, or an elegant hack condenses what you expected to be a week's worth of work into a single day or less. [1]

Prestrelené estimácie sú menej časte:

- toto je tiež problém - nevyužitý zdroj.

- positive bias
- MCE pressure
- je ťažké získať intellectually honest estimácie
- problémy ktoré nečakáme zväčša zväčšujú množstvo práce
 - Ľudia sú veľmi zlý v zohľadnení málo pravdepodobných javov.
- rozpor - exkluzívna práca na projekte vs práca

- Expert estimation
- Formal estimation model
- Combination-based estimation

Dominuje expert estimation. Formal estimation vyžaduje dôkladné požiadavky / analýzu projektu a historické data o podobných projektoch.

- Rozbiť na menšie časti : má to problémy

Functionality also not the right key. Consider a concrete example. Suppose you're building an app that logs in to a web service. Don't have individual server-side estimates for "user can create account," "account email address can be confirmed," "user can log in," "user can sign out," and "user can reset password." Have a single "user authentication" task, and estimate that. [1]

- veľa práce
- estimation fatigue

Čas potrebný na dokončenie projektu nie je možné merať v manhours.

- Brooks' law:

"Adding engineers to a late software product makes it later."

- nie všetci pracujú rovnako rýchlo; znalosť potrebnej technológie, MCE pressure, positive bias
- proxy parameter (eg. user story points, function points)

"Let's run that trail. It'll take 30 minutes." [2]

Čo má šancu fungovať?

- zmeniť plán
- pridať dobrých programátorov/expertov
- pomoc v bočných úlohách (napr. quality assurance)
- pridať ľudí na projekt včas

Ako odpovedať na žiadosť o estimáciu

XS, S, M, L, XL

- Ako sa tím učí spolu pracovať, tričká dostávajú svoj význam
- Väčšie úlohy sa môžu rozbiť na menšie
- T-shirt estimates - varovanie:
 - XL na vrchu zoznamu taskov -> nie dost' rozbité
 - SSSSS -> granularity trap
- If you have a roughly even mix of S, M, and L, you've probably structured things so that you'll have pretty good - well, least bad - estimates. [1]

- 1, 2, 3, 5, 13, 40, 100, nekonečno, nejasné
- Viditeľne odlišujú, že rozdiel medzi S a M je menší ako medzi M a L.

- Kto v diskusii dominuje je často dané osobnosťou a nie tým, kto o probléme vie viac.
- [Planing poker - Wiki](#)
- Všimnite si kartu s kávičkou. Prečo tam je?

- Vyžaduje merané data.
- Nepresné merania sú užitočné, pokiaľ sme si vedomí obmedzení.
- Kombinácia: napr. prevod story points na man-daye.

<http://www.dcs.fmph.uniba.sk/~cervenka/oose/OOSEProjectManagement.PDF>

https://en.wikipedia.org/wiki/Software_project_management

https://en.wikipedia.org/wiki/Anti-pattern#Project_management

https://en.wikipedia.org/wiki/Hofstadter's_law

https://en.wikipedia.org/wiki/Ninety-ninety_rule

[1] <http://techcrunch.com/2016/04/30/estimate-thrice-develop-once/>

http://www.ewh.ieee.org/r5/central_texas/austin_cs/presentations/2004.08.26.pdf

<http://programmers.stackexchange.com/questions/648/how-to-respond-when-you-are-asked-for-an-estimate>

https://en.wikipedia.org/wiki/Software_development_effort_estimation

https://en.wikipedia.org/wiki/Planning_poker

<https://agilefaq.wordpress.com/2007/11/13/what-is-a-story-point/>

[2] <https://www.mountaingoatsoftware.com/blog/the-main-benefit-of-story-points>

<https://rclayton.silvrback.com/software-estimation-is-a-losing-game>