

SVÄTÝ GRÁL PODPORNÝCH PROSTRIEDKOV

*Vždy, keď si slovo Problém nahradíš slovom Výzva,
dostaneš sa o krok bližšie k úspechu.*

Miroslav Soha

Slovenská technická univerzita
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava
misoha[zavináč]post[.]sk

Abstrakt. Riadenie je jedným z najpálčivejších problémov týkajúcich sa problematiky tvorby softvérových systémov. Nesprávne riadenie väčšinou vedie k predraženiu projektu, nedodržaniu termínov alebo úplnému zrušeniu projektu. Z mnohých prístupov a prostriedkov pre podporu riadenia tvorby softvérových projektov sa v tomto článku zameriame na tie, ktoré slúžia na sledovanie úloh a využitie ľudských zdrojov. Dôležitým parametrom podporných prostriedkov pre prácu v tíme je miera ich účinnosti vzhľadom na zefektívnenie práce a kvalitu výstupov tímu. Je však vždy účinkom týchto podporných prostriedkov uľahčenie práce? Hľadaním odpovede na túto otázku postupne identifikujeme výhody a nevýhody niektorých prostriedkov pre podporu práce v tíme. Súbežne porovnaním pohľadov manažérov takýchto tímov a na druhej strane členov tímu sa pokúsime načrtnúť víziu ideálneho podporného prostriedku.

Kľúčové slová: riadenie, podporné prostriedky, ľudské zdroje, Gantov diagram, softvérový projekt.

Na ceste za dokonalým softvérom

Výskum nových programových prostriedkov a technológií ide ruka v ruku s ich aplikáciou do praxe. Je to nekonečný kolobeh požiadaviek a realizácií. Na jednej strane pomyselného poľa stoja užívatelia, požadujúci aplikáciu, ktorá má nielen príťažlivý dizajn, ale zároveň obsahuje všetku funkcionality, na ktorú si dokážu spomenúť. Na druhej strane stojí tábor programátorov, testerov a analytikov, ktorí by najradšej na náprotivnej strane videli

užívateľa, ktorý chápe, že nepotrebuje vesmírnu loď na to, aby sa odviezol na nákup. Niekde uprostred tejto otvorenej partie sú projektoví manažéri, ktorí tvoria spojivo tejto zdanlivo nespojiteľnej dvojice.

Aj takto sa dá opísať dnešný stav trhu softvérových produktov pre zákazníka. Táto esej však nemá za úlohu rozoberať súvislosti a vzťahy medzi užívateľmi a programovými tímami. Chcel by som poukázať na vzťah medzi projektovým manažérom a vývojovým tímom. Ešte bližšie sa však budem venovať podporným prostriedkom pre sledovanie úloh a rozdeľovanie ľudských zdrojov v rámci softvérového projektu. Na túto problematiku sa pozrieme nielen z pohľadu projektového manažéra, ale taktiež zo strany členov vývojového tímu. Rozdielne pohľady budem demonštrovať na niekoľkých vybraných podporných prostriedkoch. V závere sa na základe odhalených odlišností v potrebách zainteresovaných strán pokúsím načrtnúť charakteristiky ideálneho a všeobecne aplikovateľného podporného prostriedku pre riadenie softvérového projektu v tíme.

Aby som podnietil čitateľovu zvedavosť, položme si hneď na úvod sériu otázok, na ktoré si skôr či neskôr v nasledujúcom texte nájdeme odpoveď. Prečo projekty potrebujú manažerov a riadenie? Sú podporné prostriedky pre riadenie a sledovanie projektu naozaj prínosom, ktorý dokáže zabezpečiť úspešné ukončenie projektu? Prečo aj napriek novým prístupom a podporným prostriedkom je taký veľký počet softvérových projektov neúspešných?

Projekt a jeho úskalia

Na softvérový projekt sa môžeme odvolávať ako na dočasnú aktivitu, ktorá je definovaná dátumom začiatku, špecifickými požiadavkami a podmienkami, určenými zodpovednosťami, rozpočtom, plánovaním a dátumom ukončenia, na ktorej sa zúčastňujú viaceré zložky. Je všeobecne známe, že tam, kde sa končí vývoj, začína údržba. Avšak vychádzajúc striktnie z definície, projekt, ktorý splnil všetky požiadavky definované zadávateľom, náklady na jeho vývoj nepresiahli rozpočet a zároveň bol dodaný načas, je považovaný za skončený.

Drvivá väčšina softvérových projektov však svojím rozsahom a zložitosťou ďaleko presahuje schopnosti jednotlivca. Dodanie takehoto projektu jednotlivcom je rádovo odhadované v rokoch, ak nie v desiatkach rokov. Schodnejšou cestou ako čakať na vytvorenie riešenia niekoľko rokov, je nasadenie vývojového tímu. Projekt sa tak môže rozdeliť na menšie celky, ktoré budú môcť riešiť súbežne viacerí členovia tímu. Tu sa však objavuje viacero problémov, ktoré sa môžu, ale nemusia prejaviť.

Podľa [7] komplikácie počas tvorby softvérového produktu v tíme súvisia najmä s obmedzeniami prirodzeného jazyka a človeka samotného. Prirodzený jazyk nie je dostatočne adresný na efektívnu a názornú komunikáciu členov tímu, prinajmenšom keď uvažíme možnosti vyjadrenia programových algoritmov, alebo vzťahov medzi softvérovými prvkami. Nemenej dôležitým faktom je, že ľudská pamäť nemá dostatočný rozsah na to, aby jednotlivec dokázal bez podporných prostriedkov sledovať aktivitu všetkých členov tímu. Navyše, ak si spomenieme na príslovie: „Koľko ľudí, toľko chutí“, príjemná predstava tímovej práce dostáva vážne trhliny. Preto je pri práci vývojových tímov na projektoch životne dôležité, aby bola stanovená ucelená koncepcia a architektúra riešenia. Toto sú hlavné dôvody, prečo je potrebné, aby bol softvérový projekt riadený.

Avšak nie je riadenie ako riadenie. Riadenie projektu bez podporných prostriedkov sa dá prirovnať k strieľaniu slepými nábojmi, môže vyzeráť atraktívne, avšak efekt je minimálny. Bez dôkladného a efektívneho riadenia riskuje každý projekt prekročenie stanoveného rozpočtu alebo lehoty dodania, čo môže viesť v lepšom prípade k nespokojnosti zákazníka, v horšom prípade k zrušeniu celého projektu. Až 31% softvérových projektov v celkovej výške takmer 81 miliárd amerických dolárov je každoročne zrušených práve kvôli neefektívnemu riadeniu[5]. Preto sa naskytá ďalšia otázka: Kde robíme chybu?

Správni ľudia na správne miesta

Základným kameňom každého projektového tímu sú ľudia a ich znalosti. Podľa analýz úspešnosti projektov v závislosti od rôznych faktorov sú ľudské zdroje jedným z limitujúcich faktorov pre úspešné ukončenie softvérového projektu. Niektoré výskumy dokonca uvádzajú, že ľudské zdroje sú jedinou voľnou premennou, od ktorej závisí úspech projektu. Táto skutočnosť poukazuje na dôležitosť správneho rozdelenia a efektívneho využitia ľudských zdrojov v rámci tvorby softvérového produktu.

Pod pojmom efektívne rozdelenie ľudských zdrojov sa ukrýva proces, ktorý má za úlohu rozvrstviť naplánované úlohy tak, aby nebol ani jeden člen tímu preťažovaný. Je samozrejmou, že rozdelenie schopností v rámci tímu nie je homogénne a rozdeľovanie úloh je podriadené schopnostiam a znalostiam jednotlivých členov. Rozdeľovanie úloh by však okrem znalostí malo zohľadňovať aj pracovné tempo a kapacitné obmedzenia pracovníkov, pretože pridelenie veľkého počtu úloh nad rámec kapacity pracovníka vedie k nárastu jeho náchylnosti k chybám a v krajnom prípade k absolútnej strate efektívneho výkonu známeho ako syndróm vyhorenia. Takýto scenár je považovaný za zlyhanie zodpovedného vedúceho tímu a dôsledkom môže byť ohrozenie úspešnosti celého projektu v závislosti od kritickosti výpadku spôsobeného pracovnou neschopnosťou postihnutého člena tímu. Projektovým manažérom a vedúcim tímov slúžia na predchádzanie takýchto stavov práve podporné prostriedky pre zobrazovanie a správu ľudských zdrojov.

Jednou zo základných techník pre zobrazovanie rozdelenia ľudských zdrojov je matica zodpovednosti. Jej použitie je zrejmé už z jej názvu. Slúži hlavne na zobrazenie rozdelenia zodpovednosti za jednotlivé úlohy respektíve podúlohy projektu. Dodefinovaním ďalších rolí pracovníkov sa nám otvára možnosť okrem zaznačenia zodpovednosti danej osoby za úlohu, definovať priamo v tabuľke aj iné role pracovníkov súvisiacich s danou úlohou. Napríklad pracovník vystupuje pri jednej etape ako programátor, ale napríklad pri ďalšej už vystupuje len ako konzultant. Výhodou tejto techniky zobrazenia je nielen jeho prehľadnosť, ale aj jeho jednoduchá škálovateľnosť pre meniace sa požiadavky rôznych projektov. Avšak musím uviesť aj fakt, že použitie tejto techniky je veľmi zložitá, prípadne ťažko realizovateľná v rámci projektov a tímov, kde sa počas realizácie úlohy menia roly pridelených členov tímu. Takáto matica by musela obsahovať doplňujúce časové alebo etapové údaje, čo by značne znížilo jej čitateľnosť.

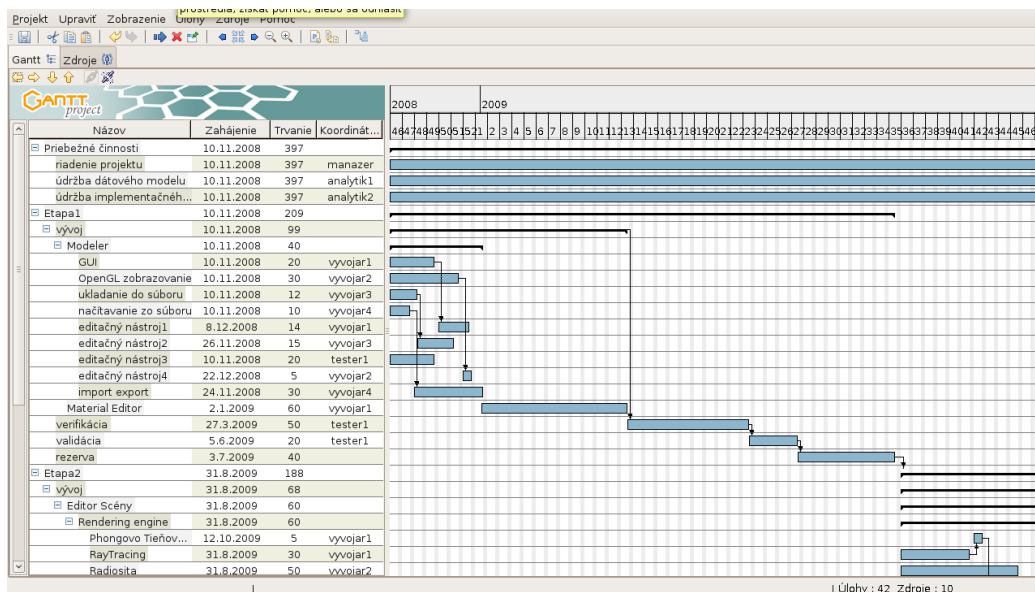
4 Miroslav Soha

Matica zodpovednosti						
	Jiří Bělohradský	Marek Borovský	Ondřej Brynda	Martin Fábry	Jakub Janeček	Zdeněk Klíma
Vytvorenie matice zodpovednosti						
Správa wiki						
Harmonogram						
Zoznámenie s Facebook API						
Správa tomcat serveru						
Analýza zdrojových súborov ankety						
Katalóg požiadaviek						
Slovník pojmov						
Use-case diagram						
Class diagram						
Stavový diagram						
Testovanie možností Facebook API						
Zoznámenie s Java servlety a JSP						
Prístup do testovacieho KOSu						

Obr. 1 – Matica zodpovednosti[1]

Použitie matice zodpovednosti je veľmi jednoduchou alternatívou pre projektových manažérov, keďže na jej vytvorenie stačí štandardný tabuľkový procesor. Ako je však možné vidieť z obrázku (pozri Obr. 1), matica zodpovednosti slúži hlavne na prehľad pracovníkov a im pridelených úloh, nenachádzajú sa tu žiadne časové informácie. Tak ako pre manažérov, tak aj pre členov vývojového tímu je veľkým prínosom možnosť disponovať takýmto prehľadom zodpovedností. Pre potreby riadenia ľudských zdrojov by bolo potrebné pridružiť ešte ďalšie podporné prostriedky.

Najznámejším zástupcom podporných prostriedkov pre riadenie založených na grafoch a schémach je Gantov diagram. V rámci plánovania a riadenia ľudských zdrojov sa využíva najmä na zobrazenie časových úsekov určených pre jednotlivých členov tímu a im pridelené úlohy. Jednoduchá úprava tohto diagramu nám môže pomôcť predísť preťažovaniu jednotlivých členov vývojového tímu. Princíp tejto úpravy spočíva v prepočte hodinovej dotácie všetkých úloh pridelených zamestnancovi v rámci daného jednotkového obdobia a porovnanie tejto hodnoty s celkovou dotáciou. Výsledok porovnania vynesieme do Gantovho diagramu podľa definovanej farebnej škály. Takto dostaneme farebne vytieňované vyťaženie jednotlivých členov tímu.



Obr. 2 – Gantov diagram[6]

Tvorba Gantovho diagramu zaberie projektovému manažérovi viac času ako napríklad matica zodpovednosti, avšak je to vyvážené prínosmi, ktoré som popísal vyššie. Nespornou výhodou pre projektových manažérov ako aj pre členov tímu je, ako vidieť z obrázka (pozri Obr. 2), intuitívne zobrazenie zadelenia času pre jednotlivých členov tímu v závislosti od definovaných úloh.

Oba príklady sú spravované a vytvárané projektovým manažérom alebo vedúcim tímu, takže ostatní členovia tímu pocítia len výhody nasadenia tohto podporného prostriedku. Preto tieto prostriedky znamenajú jednoznačné uľahčenie práce členom vývojového tímu, keďže akýkoľvek typ grafického zobrazenia zložitejších súvislostí napomáha ich pochopeniu.

Nesmieme zabudnúť na kontext, v ktorom sa zaoberáme ľudskými zdrojmi a ich riadením. Aj tá najprecíznejšie zorganizovaná skupina odborníkov by nebola schopná úspešne ukončiť projekt bez naplánovania úloh a ich následného sledovania. Preto sa teraz zameriam na podporné prostriedky pre sledovanie úloh.

Každý rád vidí úspešne ukončenú úlohu

Keď uvážime zložitosť úloh a problémov, ktorým musí vývojový tím čeliť počas realizácie projektu, je očividné, že tieto úlohy nie je možné vyriešiť bez predchádzajúcej úpravy. Na základe poznatkov zo psychológie je pre človeka najpriateľnejším postupom pri riešení zložitého problému jeho rozloženie na menšie časti, ktoré je schopný obsiahnuť svojimi znalosťami a skúsenosťami. Tento rozklad je realizovaný vo fáze analýzy a plánovania softvérového projektu. Rozdelené úlohy sú ľahšie na pochopenie a následný odhad časovej náročnosti ich realizácie, čo je výhodnou východiskovou pozíciou pre neskoršie riadenie projektu a sledovanie plnenia úloh definovaných plánom.

Dôvodom sledovania úloh nie je len monitorovanie dosiahnutých výsledkov, ale aj možná úprava plánov a riešenie problémov, ktoré vznikli počas realizácie danej úlohy. Medzi výstupmi sledovania nájdeme aj správy o priebežnej realizácii projektu, ktoré slúžia ako podklady vrcholovému manažmentu pre prijímanie strategických rozhodnutí týkajúcich sa nasledovania projektu.

Sledovanie veľkého množstva úloh a ich priebežné monitorovanie svojím rozsahom presahuje schopnosti jednej osoby. Preto boli vyvinuté moderné prístupy a podporné prostriedky kompenzujúce túto skutočnosť. Aj napriek tomuto pokroku sa môžeme aj dnes stretnúť s neadekvátnym a v niektorých prípadoch úplne zbytočným sledovaním a kontrolou v rámci softvérových projektov. Tieto situácie vznikajú najmä kvôli stavu, že neexistujú skoro žiadne štandardné postupy pre identifikovanie kľúčových vlastností, ktoré by mal projekt spĺňať pri úspešnej realizácii. Následne potom zlyháva aj odvodzenie meracích postupov z definovaných projektových cieľov, čo veľmi sťažuje rozhodovanie, ktoré merané dáta vyhodnocovať[2].

Gantove diagramy, ktoré boli spomínané v súvislosti s rozdeľovaním ľudských zdrojov môžu slúžiť aj na podporu sledovania úloh. Keďže každý záznam v Gantovom diagrame obsahuje dátum začiatku realizácie a plánovaný dátum ukončenia spolu so zodpovednou osobou, pridaním percentuálneho vyjadrenia stavu úlohy dostávame základné, prehľadné zobrazenie všetkých realizovaných úloh aj s mierou ich plnenia. Výhody takéhoto prehľadu sú aj v tom, že umožňuje zaznamenať a graficky zobrazíť časovú súslednosť jednotlivých úloh. V tejto možnosti je však ukryté aj úskalie použitia tohto prostriedku ako jediného pre sledovanie a vyhodnocovanie úloh. Spôsob zobrazenia pridelených úloh ako časovo nasledujúcich blokov zvádza používateľov k úvahám o následnosti úloh a príliš skorému plánovaniu nasledujúcich etáp. Toto sa deje na úkor identifikovania cieľov a analýzy najbližšie pridelenej úlohy, čo môže značne ohroziť úspešnú realizáciu projektu.

Názornosť Gantovho diagramu je nesporná, preto ho považujeme za prínos ako pre manažérov, tak aj pre členov vývojového tímu. Avšak, keď uvažíme jeho realizáciu tak ako bola opísaná na riadkoch vyššie, pre manažment bude nemožnosť automatického vyhodnocovania plnenia úloh znamenať značný posun v ich preferenciách. V programátorskej časti tímu priradí takejto verzii taktiež chýbajúca možnosť automatického vyhodnocovania programových zmien, čo by v praxi znamenalo posielanie všetkých úspešných zmien v zdrojovom kóde projektovému manažérovi inou cestou ako pomocou hodnotiaceho systému.

Bugzilla vznikla ako otvorený podporný nástroj pre zaznamenávanie defektov v programovom kóde. Postupne sa pridaním modulov pre správu požiadaviek, pridelovanie osôb na realizáciu, generovanie priebežných správ o stave riešenia a mnohých ďalších pretransformovala na plnohodnotný systém pre sledovanie úloh v rámci softvérového projektu. Momentálne je okrem komunity otvoreného obsahu (eng. open source) využívaná aj pri niektorých komerčných projektoch.

Projektový manažment s určitosťou privíta nasadenie podporného prostriedku takéhoto typu a to najmä pre jeho všestrannosť. Manažér má možnosť pridelovať ľudí k jednotlivým úlohám, generovať si všetky potrebné prehľady a správy a zároveň sa zníži podiel jeho administrácie na minimum. Myslím, že je to dostatočný počet dôvodov na obľúbenie si takéhoto nástroja.

U vývojového tímu však už situácia nebude taká radostná. Daňou za jeho všestrannosť je jeho vysoká zložitosť a miestami komplikované užívateľské rozhranie. Postupne je možné si zvyknúť na takéto prostredie, je však v kompetencii manažmentu, či je ochotný obetovať čas programátorov, ktorý strávia pri podpornom prostriedku na úkor kreatívnej práce na tvorbe softvérového produktu.

Svätý grál podporných prostriedkov

V úvode tejto eseje som sľúbil, že sa pokúsim načrtnúť charakteristiky ideálneho podporného produktu pre tvorbu softvérových produktov v tíme. Na niekoľkých príkladoch som poukázal na rozdiely v preferenciách medzi projektovými manažérmi a členmi vývojových tímov. Takže teraz je na rade podporný prostriedok pre budúcnosť.

Keď sa budeme riadiť pravidlom „čas sú peniaze,“ prvou charakteristikou, ktorou sa prirodzene budeme musieť zaoberať je efektívnosť práce s nástrojom. Tá závisí v prvom rade od užívateľského prostredia a možnosti upraviť si ho podľa svojich predstáv. Podľa [4] je 25% - 30% času stráveného členmi vývojových tímov na softvérových projektoch venovaných kolaborácii a komunikácii. Pre efektívne využitie členov tímu je nutné znížiť čas potrebný na ostatné administratívne úlohy znížiť na minimum. Preto je potrebné, aby mal náš produkt prehľadné a užívateľsky prívetivé prostredie.

Ak by sme zobrali do úvahy pomer objemu prezentovaných informácií aplikácie a reálneho objemu informácií, ktorý bol používateľ schopný absorbovať, ideálny stav by bol, ak by sa blížil k jednej. Túto charakteristiku je možné docieľiť upravením výstupov produktu tak, aby boli čo najnázornejšie. Preto by bolo vhodné, aby sa výsledky zobrazovali v podobe grafov, sietí alebo diagramov a tabuľky zároveň s textovými výpismi by boli použité len v zriedkavých prípadoch.

V prípade kontrolných prostriedkov akéhokoľvek druhu je potrebné sa zaoberať aj potrebou bezpečnosti vstupov, keďže chybné zadaný parameter v kontrolnom systéme môže narobiť viac škody ako objavenie sa reálneho problému počas riešenia projektu. Preto naše riešenie bude obsahovať kontrolné mechanizmy a algoritmy na kontrolu vstupov od používateľa a ich overovanie na základe už zadaných dát v systéme.

S rýchlosťou práce, ale aj s overovaním zadávaných vstupov súvisí posledná charakteristika, ktorú by som rád spomenul a to je interaktívne dopĺňanie informácií. Keď bude systém schopný zo zadaných údajov doplniť zvyšné, nie je potrebné, aby sa tým zaoberal užívateľ. Takýmto prístupom sa zmenší riziko zadania chybných údajov a taktiež sa zvýši počet operácií užívateľa za minútu a zároveň aj celková rýchlosť práce.

Týchto niekoľko charakteristík ideálneho podporného prostriedku sa postupne vynorilo počas písania tohto článku a mne ostáva dúfať, že sa vynorili aj vám pri jeho čítaní. Hľadanie univerzálneho podporného prostriedku pre riadenie softvérových systémov považujú niektorí za mrhanie časom, keď sa podobný efekt dá dosiahnuť vhodnou kombináciou už existujúcich riešení. Ja som zástancom iného názoru. Prečo používať viacero nástrojov, keď môžem mať jeden? Ale o tom niekedy inokedy.

Použitá literatura

1. Bělohradský, J. et al: Finální dokumentace Theses. Dostupné na itnernet: <https://piglet.felk.cvut.cz/courses/y36si2/theses/wiki/Final>, [cit: 2009-10-14]
2. Ciolkowski, M., Heidrich, J., Simon, F., and Radicke, M. : Empirical results from using custom-made software project control centers in industrial environments. In Proceedings of the Second ACM-IEEE international Symposium on Empirical Software Engineering and Measurement , Kaiserslautern, Germany, October 09 - 10, 2008, 243-252
3. Grover, B.: Who needs tools? [project management], In: Project Management Tools, IEE Colloquium on 18 Feb 1993, London, UK, 3/1-3/3
4. B. Helbrough, "Computer assisted collaboration - the fourth dimension of project management?" International journal of project management: the journal of the International Project Management Association, vol. 13, No. 5, pp. 329-333, 1995.
5. Romano, N.C., Jr. Fang Chen Nunamaker, J.F., Jr.: Collaborative Project Management Software, In: System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on 7-10 Jan. 2002, 233- 242
6. Vološin, M.: Môj svet. Dostupné na itnernet: <http://s.ics.upjs.sk/~volosin/swe/>, [cit: 2009-10-14]
7. Whitehead, J.: Collaboration in Software Engineering: A Roadmap. In: Future of Software Engineering, Minneapolis, MN, USA, 2007, 214-225

Annotation

The holy grail of software support tools

Management is one of the most significant parts of the software creation process. Improper management can lead up to budget overrun, passed deadlines and even the project cancellation. From plenty of different approaches and tools I will focus on the ones which are used for task tracking and allocation of human resources. Project support tool can be compared by their efficiency rates for project workers efficiency and overall project outputs. Is their effect always just efficiency? Along with others I will try to look for the answer. At last but not at least I will compare the managers' sight with the sight of developers.