

КВАЛИТЕТ СОФТВЕРА И ЊЕГОВО МЈЕРЕЊЕ

Почетке израде рачунарског софтвера током педесетих и шездесетих година карактерисала је израда "оптималних" програма од стране једне особе, једног аутора. С обзором на веома оскудне рачунарске ресурсе, аутори програма су морали да креирају програме уз пажљиво и рационално располагање тим ресурсима. Резултат тога је да су програми били права мала ремек дјела домишљатости и умијећа. Превасходни циљ је био да се добије софтвер који "ради" и да се извуче максимум из релативно скромног хардвера. Такви програми су међутим имали висок степен непоузданости у раду. Зашто? Чињеница је да се ни за један софтвер не може доказати да је исправан, него да се може само тестирати и тим показати да је највјероватније исправан, што ствара већ велику дозу ризика када је у питању поузданост софтвера. Придодајмо томе и одсуство општеприхваћене структуре програма, система рада и метода за развој софтвера. Даље, такве програме је било изузетно тешко одржавати због недостатка системског приступа у њиховој изради па према томе и због слабе документованости програма, јер је за квалитетну документацију потребан и формализован начин и језик у коме се креира документација да би била разумљива и другима. Због ових разлога одржавање програма, на примјер отклањање грешке која се појавила током рада или измјена ради унапређења програма, је било веома отежано и мукотрпно. Нарочито је то био случај када је било потребно да измјене врши и неко други, сем аутора програма, гдје се неријетко испостављало да је лакше креирати потпуно нови програм него схватити постојећи. Наравно, овакав исход би доводио, поред фрустрације, и до повећања трошкова понекад чак и за 100%.

Дакле, општа оцјена је да је овакав рад аматерски. Међутим, потреба за индустријском производњом софтвера је тражила да се у израду софтвера уведе инжењерски приступ. У овом тексту ће бити ријечи у којим активностима током израде софтвера се може примјенити инжењерски приступ, односно како је еволуирао став о могућностима његове примјене.

Циљеви који се постављају у свијету производње софтвера су: завршавање софтвера на вријеме, нулти дефект и лако одржавање.

Ако полазимо од претпоставке да се квалитет уграђује у софтвер кроз употребу процеса који у себи већ имају уграђен квалитет, онда обезбјеђење квалитета софтвера мора бити аспект активности током развоја софтвера.

Мјерења карактеристика квалитетног процеса израде софтвера

Под инжењерским приступом генерално подразумевамо: управљање, планирање, моделирање, анализирање, дизајнирање, имплементацију, тестирање и одржавање у било којој области инжењерства. Претпоставка је да се ове активности разматрају заједно са латима и техникама корисним у интеграцији ових активности.

Методолошки приступи код израде софтвера, било да су структурни, управљачки оријентисани, објектно оријентисани, оријентисани ка подацима или пак оријентисани ка формалним језицима и одговарајућим „CASE„ алатима, сами по себи не чине инжењерску дисциплину. Оно што недостаје су експерименти који се могу мјерити.

Гдје смо сада у оквиру софтверског менаџмента, када су у питању експерименти и мјерења? Показује се да често не постављамо циљеве који су мјерљиви, затим да не мјеримо елементе који чине трошкове софтверских пројеката или да не успостављамо стандарде и планове да би смо могли да обезбиједимо квалитетне софтверске технологије. Честа је и појава да многи софтверски стандарди у ствари нису стандарди, јер нису довољно прецизни, тако да адекватност не може бити одређена објективно помоћу тестова. У таквим случајевима ситуација се може третирати као да је ван контроле.

Да бисмо унаприједили и повећали квалитет мјерења у оквиру софтверског менаџмента указаћемо на аспекте мјерења који су релевантни за ову област.

Грешка која је направљена током развоја софтвера, води ка стварању дефекта у програму. Ако грешка није откривена током контроле програма, она ће проузроковати прекршаје у систему. Прекршај се може манифестовати у одступању понашања програма од захтјева током тестирања или оперативног рада. О неочекиваном понашању се извјештава као о инциденту. Неки инциденти могу бити проузроковани екстерним факторима. Ако анализом инцидента детектујемо грешку, онда ће се направити измјена у производу да би се грешка одстранила.

Кључни аспекти мјерења су: локација, вријеме, ефекти, манифестација, узроци и трошкови, и примјењују се на инциденте, прекршаје и дефекте, и промјене.

Шта то значи када су у питању, на примјер, инциденти:

Локација: адреса у програмском коду;

Вријеме: а) вријеме када се инцидент деси (интервал);

б) однос времена извршавања и тренутка дешавања инцидента (рацио однос);

Ефекти: класификација симптома (у номиналном износу);

Манифестовање: класификација према операционом профилу (у номиналном износу);

Узрок: а) грешка у дизајну;

б) грешка корисника;

ц) хардвер;

д) недијагностиковано?

Трошкови: а) сноси корисник;

- б) изгубљени ресурси;
- ц) вријеме у коме софтвер није расположив;
- д) трошкови за испоручиоца;

Узмимо сада, на примјер, промјену:

Локација: подпрограма, модул, подсистем;

Вријеме : када је промјена први пут укључена у програм/верзију;

Узрок: корекција, прилагођење, перфектно одржавање;

Примјери карактеристика квалитетног софтвера су:

Употребљивост: колико је програм једноставан и практичан за употребу;

Могућност одржавања: лакоћа са којом софтвер може бити схваћен, коригован, прилагођен и унапријеђен;

Поузданост: за коректне улазе мора да даје коректне излазе;

Портабилност: могуће је преносити софтвер са једне рачунарске платформе на другу (различити оперативни системи, различит хардвер);

Тестирање софтвера

Педесетих година се сматрало да је тестирање оно што програмери раде да би пронашли и очистили (грешке) багове у њиховим програмима. Седамдесетих година тестирање постаје процес који кроз употребу програма проналази грешке. У деведесетим годинама под тестирањем се сматра планирање, дизајнирање, формирање, одржавање и извршавање тестова и тестних окружења (тестверско инжењерство).

Међутим, поред ових достигнућа у пракси је чест случај да се тестирању посвећује мало пажње, односно процес тестирања је сиромашно дефинисан, без јасних одредница. Дакле, ситуације је често ван контроле, па је од посебног значаја да се у овој области примјени инжењерски приступ.

"Тестверско" инжењерство се бави развојем и одржавањем софтверских тест система. Такви системи треба да посједују генералност ради могућности тестирања разних софтвера. Дакле, процес тестирања мора постати аутоматизован и уведен у почетне фазе дизајнирања софтвера, и не само, као што је иначе случај, у завршну фазу, када је софтвер већ направљен.

Овим кратким приказом смо хтјели да укажемо само на неке аспекте у процесу обезбјеђења квалитета софтвера, што не значи да су то и једини. Поред ових изнесених, разматрају се и планови за обезбјеђење квалитета софтвера, вршење мануелне инспекције свих дијелова софтвера, увођење одјелења која се искључиво баве проблемом контроле процеса израде софтвера како би се унапријед редуковали трошкови које би у коначном исходу изазвале грешке.