Какво е „Качествен Програмен Код“?

В днешно време на първото нещо, за което се съблюдава при избора на продукт или услуга е качеството. Никой уважаващ себе си предприемач или домашен потребител не би закупил стока, която просто „ще свърши работа“. „Работата“ трябва да се свърши качествено! А за да бъде един продукт качествен, трябва процесът на изработка също да бъде такъв. Това важи за всички сфери на производството без изключения. Важи и за софтуерните продукти. За да бъде дадено приложение (било то десктоп или уеб-базирано), то написването му трябва да бъде качествено. Нека разгледаме по-подробно какво означава **Качествен** Програмен Код.

За писането на КПК един разработчик трябва да обръща внимание на всеки детайл от работата си. Ако трябва да погледнем „голямата картинка обаче“ качеството на кода се дели на външно и вътрешно.

Нека като за начало да уточним какво една програма да има високо външно качество. На първо място той трябва да провери дали „поведението“ на даденото приложение е такова, каквото се очаква да бъде. Едно приложение не трябва да извършва никакви други дейности, различни от тези, за които е предназначено. Следващата стъпка е да се провери дали резултатът от тези дейности е правилен – никой не си купува приложение ако то не работи „както трябва“. То трябва да бъде тествано многократно преди да бъде пуснато на пазара като се обърне внимание на специалните случаи, в които може да попадне и как се справя с тях. Разбира се, не можем да пропуснем и скоростта, с която нашето приложение ще работи. На никого не е приятно да чака дълго време за извършването на прости операции, само защото в приложението има много излишен код или такъв, който го затормозява без да е необходимо. Още нещо, на което се обръща внимание е потребителският интерфейс на приложението – удобен ли е за употреба и „гали“ ли той окото. Колкото и да е добро нашето приложение ако за крайният потребител е трудно да го използва без да има специализирани познания, то едва ли ще се радва на висока популярност сред клиентите. Много важна част от правна и етична гледна точка е сигурността на нашата програма. Със сигурност не е добра идея всеки да има достъп до нашето приложение и да може да прави промени в него. Да не говорим за разкриването на лични данни – това е **абсолютно** немислимо. Затова нашето приложение трябва да е сигурно – всички данни да бъдат енкапсулирани и да не могат да бъдат достъпвани от трети лица, без разрешение.

Всичко това се отнася към външното качество на кода. Можем да предположим, че всички (добри) програмисти се стараят техните приложения да отговарят на тези изисквания. Но много от тях, съзнателно или не, не следят за вътрешното качество на кода. То се има отношение най-вече към разработчиците – тези, които пишат, поддържат и доразвиват приложението. Най-фундаменталният въпрос, с който можем да започнем при проверката на вътрешното качество е „Лесен ли е кодът ни за четене и разбиране?“. Не можем да очакваме едно приложение да постигне успех, ако не предлагаме към него и добра поддръжка. А за осъществяването на добра поддръжка е необходимо да екипите, отговарящи за нея, да прекарат доста време в четене и разбиране на кода. Това време може да бъде редуцирано ако кодът е написан правилно – на всички структури, класове, методи, променливи и т.н. са дадени смислени имена и е сравнително лесно да се проследи логиката на кода. Като заговорихме за логика е добре да споменем, че всяка програма трябва да извършва своята дейност, следвайки възможно най-проста логика. Никой не би имал полза ако приложението е написано сложно, карайки четящият да влага усилия да разбере всеки един ред. Това ни води и до следващата проверка, а именно – дали кодът е добре структуриран. Иначе казано – спазени ли са официалните конвенции и тези на организацията (ако има такива) при писането на код. Най-добрите примери в тази връзка са „силната кохезия“ и „слабата зависимост“. Всеки клас/метод/променлива трябва да изпълнява конкретни функции и да бъде колкото се може по-малко зависим от други такива. Разбира се съществуват и много други основни правила при писането на код като например правилното описване на класовете и всички техни полета, конструктори, свойства и методи – те трябва да са в определен ред и да отговарят на различни изисквания свързани с форматирането и конкретността на функциите, които трябва да изпълняват. По този начин четящият много само с един поглед може веднага да разбере за какво се отнася кодът, който чете. Не на последно място трябва да се обърне внимание на това до колко е възможно модифицирането на дадена програма в бъдеще. Едва ли има приложение на света, което да не е претърпяло промени във времето. И не би трябвало. Всички добри приложения постоянно се разрастват, придобиват нови функционалности и същевременно стават все по-лесни за употреба. Затова е много важно програмистите да си дават възможността един на друг да развиват приложенията, пишейки преизползваем код, създавайки добра структурна йерархия и т.н.. Понякога обаче, колкото и да добре да е написан кодът в програмата, той не може да бъде напълно разбран. Тогава на помощ идва документацията – можем ясно и детайлно (ако е наложително) да обясним какво и защо прави дадения клас/метод и т.н. Важно е да се акцентира на въпросите **какво** и **защо**, а не **как**. При писането на коментари много програмисти описват как работи техният код, което в повечето случаи е напълно излишно, а понякога объркващо и затормозяващо.

Със сигурност има още много неща, за които трябва един програмист да внимава при писането на апликация. Описаните по-горе обаче са фундаментални и не трябва да бъдат пренебрегвани при никакви обстоятелства ако претендираме, че произвеждаме качествени продукти.