QA Технология и възможностите за аутсорсинг

Автор Таня Миткова за Ениак ООД

Всеки софтуерен продукт се разработва по определена методология, преминава през различни етапи и фази на разработка, преди да достигне до потребителите. В този документ ще представим два от методите - Waterfall и Evolutionary и ще направим кратко сравнение между тях. Описваме етапите на разработка на софтуерния продукт - Проект, Alpha и Beta софтуер, Финален софтуер и Реализация и ще се постараем да опишем ролята на програмистите и тестерите в тях. В допълнение, целта ни е да покажем как могат да се извършат от трета независима организация част от софтуерните тестове. Постарали сме се да представим нашето разбиране, опита ни в досегашни проекти и ресурсите на Ениак за изпълнение на тестовете.

Методи за разработка на софтуерни продукти

При започването на една софтуерна разработка се представя списък с етапите на проекта, сроковете за изпълнение и нужните ресурси. Планирането на действията зависи и от метода за разработка на софтуерния продукт - дали сме избрали традиционния Waterfall Method или алтернативния Evolutionary Method.

Основните характеристики на тези методи са:

Waterfall Method - Това е класическия модел за разработка на софтеурни проекти. При него се преминава последователно през анализ на изискванията, изработка на спецификациите, програмиране, тестване, внедряване. Характерно за този модел е, че един етап е почти завършен, преди да се премине към следващия, т.е. функционалните изисквания са завършени и се подготвят спецификациите, програмирането започва след като функционалните спецификации са напълно завършени, при тестването екипа разполага с пълната документация по проекта в началото на етапа тестване. Този метод е подходящ за проекти изработвани по поръчка на клиента. Той позволява клиента да определи в началото каква функционалност очаква от софтуерния продукт и цената на проекта е по-лесна за определяне. При Waterfall Method не се допускат големи промени, след като изискванията и спецификациите са одобрени от двата партньори - клиент и разработчик.

Добрата страна на този метод е възможноста да се планират ресурсите, времето и средствата още в началото на проекта. Рискът при използване на този метод е в ранното вземане на решения за функционалността на системата. В първия етап има представи и прототипи на бъдещия продукт и е възможно в процеса на разработка да се наложи промяна на идеите. Някои части могат да се окажат много трудни за програмиране и тестване, не са достътъчно познати конкурентните продукти.

Evolutionary Method - Този метод се характеризира с добавянето на нови функции към ядрото на софтуерния продукт. Ядрото е нужно да бъде проектирано така, че да осигурява, от една страна лекота при добавянето на новите функции и да е достатъчно стабилно и съдържащо базовата функционалност. На всеки етап от развитието на продукта се добавят нови функции към ядрото. В началото програмистите разработват ядрото, като включват само няколко свойства в него. Започва пълно тестването на първоначалната версия, сякаш тя е финален продукт, тестовете и отстраняването на грешките продължава, докато ядрото се превърне в достатъчно стабилен продукт. Тогава програмистите прибавят малка група от нови функции и тест-екипа извършва нов кръг от тестове, разработчиците отстраняват грешките до стабилизиране на продукта. Развитието на софтуерния продукт продължава с прибавяне на следващите няколко нови функции към

системата, последвани от етап на тестване и изчистване на пропуските до създаването на първоначалния продукт, готов да бъде представен на клиентите.

Предимствата на този метод са във възможността по-лесно да се добавят нови функции, или да се променят тези, които са се оказали неефективни при предходните версии. Продуктът преминава през няколко тест-фази и е по-лесно да се определи времето, нужно за тестване и да се фиксира краен срок за изпълнение на целия проект. При Waterfall Method има несигурност, породена от тестването в "последния" момент. Никой не знае колко грешки ще се открият, колко време ще е нужно за откриването и отстраняването им. При проектите разработвани по Evolutionary Method, поради множеството тест-цикли е възможно доста по-рано да се направят предвижданията за всички тези параметри.

Някои от рискове при използването на Evolutionary метод са програмистите да добавят нов програмен код към ядрото, преди да е завършило тестването на предишната версия, или преди да са отстранени старите грешки. При тези проекти тестването трябва да започне на много ранен етап и да се премине през пълен тест цикъл на ядрото. Понякога тест-лидера не разполага с достатъчно свободни тестери и тогава в ядрото остават много неоткрити грешки. Това е в противоречие с принципите на метода за стабилно ядро и добавяне на нови функции към него и ще причини множество проблеми при приближаване към финалните за проекта срокове.

Кратко сравнение на двата метода

Има много статии и книги, които определят Evolutionary Method като по-съвременния метод за ръководене на проекти и разработка на софтуерни продукти. Голямото предимство на този метод е в обратната връзка с клиентите, в мобилността и лесното добавяне на нови функции. Този метод се използва при проекти с неограничен брой крайни клиенти - фирмата разработва софтуерен продукт и очаква той да бъде закупен от 5000 или 20000 независими клиенти, с увеличаването на броя на клиентите се увеличава ефективността (приходите) от проекта. Тези непознати крайни клиенти са хората, които ще платят за продукта и ако направим сравнение с Waterfall Method, можем да кажем, че те са клиентите, които поръчват изработката на софтуера. Не е възможно да знаем техните нужди и желание в първия етап на проектирането. Изискванията се променят във времето, има нужда от нови вересии на продукта и обратната връзка с клиентите дава новите изисквания, по които се разработват спецификациите. Този метод се прилага в някой от най-големите компании от IT индустрията - IBM, HP, Microsoft.

Изборът на метод зависи от същността на софтуерния продукт, който предстои да се разработва. Waterfall Method е по-подходящ за разработка на индустриални системи и такива, пряко свързани с производството. Вие не бихте искали да използвате автомобил, в който електрониката за управление е разработена по Evolutionary Method, или да летите с авиокомпания, работеща по стандарт различен от CMM 5. При тези проекти едно от основните изисквания е надеждността, допълнена с разработването на всички възможни ситуации, преди продукта да е влязъл в експлоатация, и Waterfall Method е най-използвания.

Нашият тест подход при двата вида методи

Нашата подготовка, нагласа и очаквания за софтуерните тестовете са различни при Waterfall Method и при Evolutionary Method. При Waterfall Method сме подготвени за тест-тур, които е с голяма продължителност и е концентриран в етапа на Alpha и Beta фазата. При

Evolutionary Method участваме с много на брой, но по-кратки тест фази, имаме много повече етапи на Regression Test, анализираме внимателно релацията между новодобавената функция и ядрото, така че да можем по-бързо и ефективно да определим областите с възможност за възникване на проблеми. Можем да организираме и група от beta-тестери, и да използваме техния подход и отношения към продукта, за да извършваме по-ефективно тестовете и да ви предоставим обратна връзка на един ранен етап. При определянето на екипа от тестери се стремим да привлечем и тестер, който е много близо до индустрията, за която е предназначен софтуерния продукт - например, ако вашия продукт ще се използва от адвокати, ще се постараем да привлечен в екипа и тестер с опит и интереси в областта на правото. От него няма да очакваме да бъде най-добрия или основен тестер по проекта. Той ще работи под ръководството на някой от по-опитните тестери и ще очакваме да ни помогне да разберем по-добре индустрията и бъдещите ви клиенти.

Етапи при разработка на софтуерен продукт

Независимо от избрания метод за разработка, всеки продукт преминава през няколко етапи.

a. Проект на софтуерния продукт
b. "Alpha" софтуер
c. "Beta" софтуер
d. Финален софтуер
e. Реализация

Възможно е и по-детайлно описание на етапите. Тук ще приемем тези етапи, като най-често дефинираните в процеса на разработка на софтуерен продукт.

a. Проект на софтуерния продукт

Това е етапа, в който се определят изискванията, спецификациите, прототипите. Програмирането започва в края на този етап. Тестването в етапа на проект е свързано с проверка на документите, а не с проверка на програмен код. Ръководителят на тест екипа анализира и подготвя различни видове данни, преценява нуждите от допълнително оборудване, обмисля какви програмни продукти са нужни за извършване на тестовете, може да започне да установява контакти с бета-тестери.

Участието на Ениак като външна тест организация

На този етап ние в Ениак можем да ви предложим преглед на пълнотата на потребителския интерфейс. Можем заедно да анализираме данните и методите за тестване на бъдещия продукт. Ние ще бъдем подготвени за следващите тест етапи и ще можем да планираме ресурсите и тестерите, съобразявайки се с вашия проект и нужди.

b. "Alpha" софтуер

Това е един "исторически" етап в процеса на разработка. Има много определения за това, кога един продукт е в "Alpha" фаза. Най-често се приема, че продукта съдържа всички функции, но някои от тях е възможно да не са завършени напълно или да имат грешки. В тази фаза продукта може да покаже своята същност и може да се използва за работа. Това е етапа, в който се включват най-много тестове. Тестват се всички области на системно и функционално ниво, тестовете са много детайлни и се изпълняват в строга последователност. Описват се намерените грешки. Програмистите ги отстраняват и започват етапите на тестване за да се удостовери, че поправената грешка, не е причинила проблеми в други области от програмата (Regression Testing).

Участието на Ениак като външна организация в "Alpha" фазата

Това е най-важния етап от разработката и тестването на един продукт и нашето участие може да бъде на различни нива, в зависимост от вашите нужди. Има два основни начина за използване на нашите тестерски услуги:

- Основни тестери по проекта
- Да допълним вашия екип от тестери

Когато сме **основни тестери по проекта**, ние сме отговорни за изработката на тест стратегията, тест плана, тест сценариите. Най-доброто време за включването ни в проекта е на етапа на дизайн или преди "Alpha".

Изработка на Тест План

Изработката на тест плана е една от най-съществените части от процеса на тестване. В тест плана ще опишем функциите, които трябва да се тестват, подхода, допусканията, критериите за оценка на резултатите, нужните ресурси за извършване на тестовете.

Подходът при тестването има някои особености, в зависимост от избрания метод за разработка на софтуерния продукт. Ако вие разработвате продукта по Waterfall Method, ние ще формулираме нашата тест стратегия като използваме вашата документация и спецификации към проекта. Те обикновенно са обстойни и детайлни и ние ще извършим нужните тестове, за да удостоверим, че финалния продукт отговаря на първоначално дефинираните функционални изисквания. Ако разработвате продукта по Evolutionary Method или по някой друг от agile методите, ние ще изградим нашата тест стратегия като анализираме изискванията от вашите спецификации, бизнес правила за вашия бранш, ръководството на потребителя, разговори с реални клиенти. Нашия подход включва съвместна работа по изработката на тест плана и извършване на част от тестовете. Когато работим по изработката на тест плана, ние отделяме няколко часа, за да изпробваме нашите идеи и да работим с продукта. По този начин научаваме много за него, тестваме, спестяваме време и намираме част от грешките. Това ще помогне да определим, кои части от системата е нужно да се тестват, местата с най-голяма вероятност за възникване на проблем, какви тестове да разработим и приложим. В допълнение винаги ще мислим и като реални потребители на софтуера, така че вие ще получите и обратна връзка с бъдещите си клиенти. Вета-тестерите при Evolutionary Method е нужно да се включат на по-ранен етап и с по-голямо внимание да се следи техното поведение и отношение към продукта.

Когато сме **допълнителни тестери** по проекта, ние обикновено използваме тест план, предоставен от вашия екип, или изпълняваме вашите указания и изисквания за провеждане на тестовете. Най-доброто време за включването ни в изпълнението на тестовете е в "Alpha"фазата.

Оборудване на тест лабораторията

Ще конфигурираме нужните за тестовете устройства, определени в тест плана. Използваме оборудване на нашата лаборатория, специализирано оборудване на клиента и ако е необходимо допълнителни устройства. Нужно е да установим система за отчитане на намерените грешки - различни видове отчети и форми. По този начин се установява добра отчетност по проекта в посока намерени, отстранени, чакащи грешки.

Изпълнение на тестовете

Всички предварително проектирани и проверени тестове се изпълняват в нашата лаболатория. Когато резултатите от тестовете са различни от очакваните се изготвя отчет с описание на ситуацията и се въвежда в системата за отчитане на грешките. Тази система може да бъде on-line и вие ще можете да видите веднага новите спорни въпроси. Другата възможност е да изпращаме по е-майл отчетите с резултатите от тестовете.

Описание на грешките

Ние знаем, че описанието на грешките е много важен етап от тестването. То трябва да е в писмен вид, просто, разбираемо, с възможност да се възпроизведе от друг член на екипа. Нашите отчети включват - версия на продукта, дата, вид на грешката, сериозност, име на проблема, описание и стъпки за възпроизвеждането му, име на тестера, допълнителна информация. Отделяме специално внимание на описанието на проблема и начина за неговото възпроизвеждане. Ще ви дадем обосновка защо мислим, че това е проблем и начина, по който сме достигнали до него. Вземайки под внимание факта, че при нашите проекти програмистите и тестерите обикновено се намират в различни офиси и е по-трудно да отговаряме на въпроси, ние използваме следния подход, за да се уверим, че сме описали ясно грешките - След като един тестер подготви отчета, друг тестер се опитва да възпроизведе проблема само по описанието му. Всеки отчет се ретества внимателно, преди да го въведем официално в системата за отчетане на грешки, ако е нужно разработваме нов случай, където се вижда по-ясно проблема. Това е наша методология за описание на проблема, която сме развили и използвали в различни тест проекти.

c. "Beta" софтуер

Когато един софтуер е в Beta етап, това обикновено означава, че той е готов да бъде показан на крайните потребители за оценка, но не винаги означава, че продукта е напълно завършен. Възможно е да има няколко сериозни грешки и да се работи за отстраняването им. Определянето на Beta етапа зависи и от избрания метод за разработка.

При Waterfall Method, **Beta етапа** означава, че всички детайли са програмирани и тествани, документацията за крайния потребител е почти 50% завършена и не се очакват сериозни нови задачи за програмистите.

При Evolutionary Method, **Beta етапа** означава, че ядрото на продукта е завършено, всички необходими за минималната функционалност свойства са програмирани, но е възможно нови, привлекателни функции да се добавят. Beta етапа се достига много рано при този метод.

Независимо от избрания метод на разработка в този етап се включват Beta тестери - това са подбрани крайни потребители, които не работят във вашата организация, но ще използват софтуерния продукт в бъдещите си дейности и ще ви разкажат за своето мнение и опит с продукта. Темата за Beta тестерите е много интересна за нашия екип и има доста статии по въпроса за ефективното използване на тяхното мнение.

Участието на Ениак като външна организация в "Beta" фазата

- от Alpha към Beta

Когато сме започнали работата с вашата организация през Alpha етапа, през Beta се ретестват бързо всички поправени грешки, пълно се тестват всички модули, който ще работят съвместно, анализира се и се търсят, на базата на отстранените грешки, нови области, където е възможно програмата да не е достатъчно стабилна. През този етап ние познаваме добре софтуера и знаем какви са очакванията за неговата функционалност. Това е времето, в което влизаме в ролята на крайни потребители. Например, ако тестваме софтуер за on-line поръчки на компютърни компоненти, ние правим, регистрация, избираме продукти, променяме поръчката, потвърждаваме промените и очакваме системата да отговори адекватно на всички наши заявки. В книгата "Testing Computer Software" (authors Cem Kaner, Jack Falk, Hung Quoc Nguyen; year of publication 1999) се описва този начин на тестване в етапа Beta. Той е изцяло различен от Функционалните тестове, проведени през предишните етапи и позволява да се открият доста нови грешки. И в допълнение, това е един етап, които ние много харесваме, защото ни дава свобода да работим с продукта и можем да използваме цялата си фантазия за генериране на различни извънредни случаи.

d. Финален софтуер

След завършване на Beta етапа, продукта навлиза във фазата на последна оценка и преглед на съпътстващите материали, преди да се представи пред потребителите. Приема се, че всички грешки са отстранени, възможно е за някои да е взето решение да бъдат отстранени със следващата версия. Важен е въпроса за оценка на надежността. Известен е факта, че няма софтуер без грешки и винаги е възможно някои части да се направят по-добре. Оценката на надеждността е комплексно решение на ръководството, при него трябва да се балансира между риска и цената. Ролята на тестера в този етап е да покаже какви са рисковете в следващите няколко месеца и по-този начин да подпомогне ръководството при вземането на решението.

Финален тест

Към този тест се преминава, след като всички грешки са отстранени, резултатите от Beta теста са взети под внимание, не се предвиждат никакви видими промени в интерфейса. Тест екипа изпълняват една последна вълна от сценарии, като се държи като потребител, който ще работи с продукта. Много често, през този етап се използват услугите на външна тест организация за допълнително осигуряване, преди взимане на крайното решение за степента на завършеност на софтуера.

Използвана литература:

"Testing Computer Software", Cem Kaner, Jack Falk, Hung Quoc Nguyen, 1999