Софтуерно Инженерство

1.Основи на процесът на разработка на софтуер

Софтуерното инженерство (на английски: Software engineering) е проектирането, прилагането и видоизменянето на софтуер с цел неговото високо качество, приемлива цена, поддръжка и бързо разработване. Това е „систематичен подход към анализа, проектирането, оценяването, прилагането, тестването, поддръжката и повторната разработка на софтуер, или с други думи приложение на инженерната наука към софтуера.

Разработката на софтуер (на английски: Software development, също като разработване на приложен софтуер, софтуерен дизайн или проектиране на софтуер, софтуерно програмиране, създаване и писане на софтуерна документация, тестване на софтуер и поправяне на софтуерни грешки) е бизнес процеса на писане на програмен код, неговата поддръжка, но в по-широк смисъл включва всичко, което стои между първоначалната концепция за определена програма или желан софтуерен продукт до релийза, понякога това става като планиран процес на разработка. Разработката на софтуер може да включва изследвания, нови разработки, създаване на прототипи, модификация, повторно използване, ре-инженеринг, поддръжка и всякакви други дейности, чийто краен резултат е софтуерен продукт.

При разработката на програмния продукт програмираното му е съобразено с нуждите на дадена целева потребителска група, която се интересува от създаването или подобряването на такъв софтуерен продукт и съответства на маркетинга му. Но софтуерът може да бъде разработван и по множество причини, като най-общо трите са да отговаря на конкретните нужди на потребителите, на дадена фирма или клиент (на английски: custom software или bespoke software), да отговаря на възприетите нужди на група от потенциални потребители (често софтуер с отворен код или рекламен) или дори за лична употреба (например напредналият програмист или изследовател-учен може да напише програма за автоматизиране на сложни задачи). Понякога се налага разработката и на вграден софтуер (асемблиране), например когато се изисква процесът на разработка да бъде интегриран с разработката на контролиран физически продукт. Системният софтуер засяга приложните програми и самия процес на програмиране, поради което често се разработва отделно.

Нуждата от по-добро качество на процеса на софтуерна разработка води до началото на софтуерното инженерство, което се стреми да приложи систематичния подход, илюстриран в инженерната парадигма, към разработката на софтуер.

Съществуват много подходи към управлението на софтуерни проекти, известни като циклични модели в живота на софтуерната разработка, методологии, процеси или модели. „Моделът на водопада“ на английски: waterfall е традиционният подход, а т.нар. гъвкава методология на английски: agile разработка на софтуер е по-модерен подход.

2. Етапи в разработката на софтуер

Планиране: Анализът на планирането и изискването е началният, жизненоважен и основен етап в SDLC, който се изпълнява от старшите членове на екипа с принос от клиенти, отдел продажби, експерти в областта на домейните, пазарни проучвания в бранша. След това цялата тази информация се натрупва и се използва за планиране на основния подход към проекта, за провеждане на проучване за осъществимост на продукта по всички начини от икономически, оперативни до технически области. Проучването за осъществимост отваря много врати за изпълнение на проекта, но подходът се решава въз основа на най-високия фактор на качество с минимален риск.

Дефиниране: Анализът след изискване, дефинирането и документирането на изискванията по ясен начин е много важно. След като това е направено, тези документирани изисквания трябва да бъдат одобрени или от клиенти, или от пазарни анализатори. SRS (спецификация на софтуерните изисквания) влиза в картина за това, която съдържа всички изисквания на продукта, които трябва да бъдат проектирани и разработени по време на жизнения цикъл на проекта.

Проектиране: SRS се счита за ориентир за продуктови архитекти, за да предложи най-добрата архитектура за продукта, който ще бъде разработен. Въз основа на обхванатите изисквания, за архитектурата на продукта се предлага повече от един подход. Всичко това е документирано в DDS (Спецификация на документа за проектиране). Тогава DDS се преглежда и анализира от важни членове / заинтересовани страни въз основа на различни фактори, като пазарни рискове, модулна конструкция, бюджетни ограничения, ограничения във времето, надеждност на продукта, най-добрият подход за проектиране и т.н. Подходът на дизайна носи цялата информация. За архитектурния модул на продукта под формата на диаграма на потока от данни. Вътрешни или външни, най-малките подробности трябва да присъстват в документа за проектиране.

Изграждане: Това е етапът, в който започва реалното развитие на продукта и се изгражда продуктът. Позовавайки се на диаграмата на потока от данни в документа за проектиране, програмният код се генерира. Колкото по-ясна е предназначена за документиране, толкова по-безпроблемно е генерирането на код. Кодерите / разработчиците трябва да следват указанията за кодиране, дефинирани от инструментите им за организация и програмиране, като интерпретатори, компилатори, отстраняване на грешки и т.н., за да направят кода по-четим и ясен.

Тестване: Тестването е част от повечето от всички етапи в SDLC, следователно можем да кажем, че тестването е подмножество от всички етапи. Тестването е етап, в който дефектите на продуктите се отчитат, проследяват, фиксират и проверяват отново, докато продуктът не достигне определените стандарти за качество съгласно SRS.

Какво е разработка на софтуер?

Разработването на софтуер е итеративен логически процес на разработване на индивидуален софтуер, използващ специфичен език за програмиране. Този процес би могъл да се раздвои в създаване, уточняване, проектиране, програмиране, документиране, тестване, отстраняване на грешки.

Това може също да включва изследвания, прототипиране, модификация, реинженеринг и повторно използване. Разработването на серия от кодове с взаимосвързани функционалности се използва за разработване на софтуер.

3. Методологии за разработка на софтуер

Какви са методологиите за разработване и тестване на софтуер?

Тестването е съществена част от процеса на разработване на софтуер. Стабилен и стабилен софтуерен продукт може да бъде доставен с използване на стандартни методологии за тестване, които ще помогнат за прогнозиране на времевата линия на софтуерната система.

Софтуерното приложение може да стане още по-сложно с голям брой платформи и устройства. По-важното е, че се изисква да се гарантира дали те отговарят на посочените изисквания и могат ли да бъдат ефективно инсталирани и експлоатирани на машината на потребителя или не.

Значение на методологиите за тестване

Методологиите могат да се разглеждат като набор от механизми за тестване, използвани в жизнения цикъл на разработка на софтуер от Unit Testing до System Testing. Изборът на подходяща методология за тестване се счита за ядрото на процеса на тестване.

4. Софтуерни изисквания

Софтуерни изисквания е поле отвътре софтуерно инженерство който се занимава с установяване на нуждите на заинтересованите страни, които трябва да бъдат решени чрез софтуер. Стандартният речник на IEEE на терминологията на софтуерното инженерство определя a изискване като

Условие или способност, необходими на потребителя за решаване на проблем или постигане на цел.

Условие или способност, които трябва да бъдат изпълнени или притежавани от системата или системния компонент, за да се изпълни договор, стандарт, спецификация или друг официално наложен документ.

Документирано представяне на състояние или способност, както е в 1 или 2.

Дейностите, свързани с работата със софтуерни изисквания, могат да бъдат разделени на извличане, анализ, спецификация и управление

Основна статия: Спецификация на софтуерните изисквания

Спецификацията включва представяне и съхраняване на събраните знания по постоянен и добре организиран начин, който улеснява ефективната комуникация и управлението на промените. Случаи на употреба, потребителски истории, функционални изисквания и модели за визуален анализ са популярен избор за спецификация на изискванията.

Както можете да видите, Анализът на изискванията е първата дейност в SDLC, последвана от Функционална спецификация и т.н. Анализът на изискванията е жизненоважна стъпка в SDLC, тъй като резонира с тестове за приемане, които са критични за приемането на продукта от клиентите.

В този урок ще обясним как се прави анализ на изискванията в SDLC. Също така ще видим различните участващи стъпки, резултати, предизвикателства и коригиращи мерки при анализа на изискванията.