Курсова работа

1. **Софтуерна документация** е текст, който придружава компютърен софтуер или е вградена в изходния код. Документацията обяснява как функционира софтуерът или как да се използва и може да означава различни неща за хората в различни роли. Документацията е важна част от софтуерното инженерство.
2. **Видовете документация включват:**

* ***Изисквания***- Изявления, които идентифицират атрибути, възможности, характеристики или качества на системата.
* ***Архитектура / Дизайн*** - Преглед на софтуера. Включва връзки с околната среда и принципи, които да се използват при проектирането на софтуерни компоненти.
* ***Технически*** - Документиране на код, алгоритми, интерфейси и API.
* ***Краен потребител*** - Ръководства за крайния потребител, системните администратори и помощния персонал.
* ***Маркетинг*** - Как да продаваме продукта и анализ на пазарното търсене.

1. **Документацията за изискванията** е описанието на това, което даден софтуер прави или трябва да направи. Използва се по време на разработката. Изискванията се използват от всички, участващи в производството на софтуер: крайни потребители, клиенти, дизайнери, разработчици и тестери. Изискванията се предлагат в различни стилове, обозначения и формалност. Могат да бъдат целеви, близо до дизайна, могат да бъдат посочени като твърдения на естествен език, като изчертани фигури, като подробни математически формули и като комбинация от всички тях. Изискванията може да са неявни и трудно да бъдат разкрити. Необходимостта от документация за изискванията обикновено е свързана със сложността на продукта, въздействието на продукта и продължителността на живота на софтуера.

* Ако софтуерът е много сложен или разработен от много хора, изискванията могат да помогнат за по-добрата комуникация какво да се постигне. Ако се очаква софтуерът да работи само месец или два, може да е необходима много ограничена документация за изискванията.
* Ако софтуерът е от критично значение за безопасността и може да има отрицателно въздействие върху човешкия живот, често се изисква по-официална документация за изискванията.
* Ако софтуерът е първа версия, която е надградена по-късно, документацията за изискванията е много полезна, когато управлявате промяната на софтуера и проверявате, че нищо не е нарушено в софтуера, когато е модифициран.

1. **Архитектурната документация** (софтуерната архитектура) е специален тип проектни документи. Много важна част от документа за проектиране при разработването на софтуер за предприятието е документът за проектиране на база данни. Той съдържа концептуални, логически и физически елементи на дизайна. Включва официалната информация, от която се нуждаят хората, които взаимодействат с базата данни. Целта на подготовката му е да се създаде общ източник, който да се използва от всички.
2. **Потенциалните потребители са:**

* Дизайнер на база данни
* Разработчик на база данни
* Администратор на база данни
* Дизайнер на приложения
* Разработчик на приложения

1. **Когато говорим за релационни бази данни, документът трябва да включва следната информация:**

* Таблици,
* атрибути и техните свойства,
* изгледи,
* ограничения като първични ключове, външни ключове.

1. **Техническа документация**

Важно е кодовите документи, свързани с изходния код (които могат да включват README файлове и API документация), да бъдат задълбочени, но не толкова многословни, че да отнемат твърде много време или да се поддържат трудно. Обикновено се намират различни ръководства за документация, специфични за софтуерното приложение или софтуерния продукт, документирани от API авторите. Тази документация може да се използва от разработчици, тестери, а също и крайни потребители. Кодовите документи често са организирани в справочно ръководство - стил, позволяващ на програмист бързо да търси произволна функция или клас.

**1.Техническа документация, вградена в изходния код**

Често инструменти като Doxygen, NDoc, Visual Expert, Javadoc, JSDoc, EiffelStudio, Sandcastle, ROBODoc, POD, TwinText или Universal Report могат да се използват за автоматично генериране на кодовите документи - тоест те извличат коментарите , където е възможно, от изходния код и създават справочни ръководства като текстови или HTML файлове. Идеята за автоматично генериране на документация е привлекателна за програмистите по различни причини. Например, тъй като е извлечен от самия изходен код (например чрез коментари), програмистът може да го напише, докато се позовава на кода, и да използва същите инструменти, използвани за създаване на изходния код, за да направи документацията. Това значително улеснява поддържането на документацията актуална. Разбира се, недостатъкът е, че само програмистите могат да редактират този вид документация и от тях зависи да я опресняват.

1. **Грамотно програмиране**

Уважаваният компютърен учен Доналд Кнут отбеляза, че документацията може да бъде много труден процес на последващо обмисляне и се застъпи за грамотно програмиране, написано едновременно и на място като изходния код и извлечено с автоматични средства. Езиците за програмиране Haskell и CoffeeScript имат вградена поддръжка за проста форма на грамотно програмиране, но тази поддръжка не се използва широко.

1. **Изяснително програмиране**

То е резултат от практически приложения на грамотното програмиране в реални контексти на програмиране. Парадигмата Elucidative предлага изходният код и документацията да се съхраняват отделно. Често разработчиците на софтуер трябва да могат да създават и да имат достъп до информация, която няма да бъде част от самия изходен файл. Такива пояснения обикновено са част от няколко дейности за разработка на софтуер, където изходният код на трета страна се анализира по функционален начин.

1. **Потребителска документация**
2. **Маркетингова документация**

За много приложения е необходимо да има някои рекламни материали, които да насърчават случайните наблюдатели да отделят повече време за изучаване на продукта.

**1.Тази форма на документация има три цели:**

* Да развълнува потенциалния потребител за продукта и да му внуши желание да се ангажират по-активно с него.
* Да ги информира какво точно прави продуктът, така че очакванията им да са в съответствие с това, което ще получат.
* Да се ​​обясни позицията на този продукт по отношение на други алтернативи.