

11: The Engineering Process - Резюме

Софт. инженерство е систематичен и дисциплиниран подход за проектиране, разработка, поддръжка и тестване на софт. продукти.

Жизнен цикъл на софтуера (SDLC): всички етапи от разработката на софтуера – от анализа на изискванията до поддръжката.

Ключови етапи: анализ, дизайн, имплементация, интеграция, тестване и пакетиране.

Критики към традиционния подход: Често разработката е хаотична, известна като „**code and fix**“. При липса на ясно планиране и дизайнът системата страда, което води до трудности при добавяне на нови функции и отстраняване на грешки.

Софтуерен процес: рамка от дейности, действия и задачи за създаване на качествен софтуер навреме. Who, What, When, How

Жизнен цикъл на разработка на софтуер (SDLC):

- процес за проектиране, разработка и тестване на качествен софтуер

- целта е да се произведе софтуер, който отговаря на очакванията на клиентите и се завършва в рамките на определеното време и бюджет

- ISO/IEC 12207 е международен стандарт за процесите на жизнения цикъл на софтуера.

- Фази:** Планир., Дефиниране, Дизайн, Build., Тестване, Deployment

Unified Process (UP): итеративна и инкрементална рамка за гъвкаво управление на проектите, която е ориентирана към случаи на използване (use-case-driven) и архитектура.

Фази на софтуерната разработка:

1/Анализ:

- Анализ на изискванията: дефиниране на функционалностите.
- Домейнов анализ: моделиране на концепции и взаимоотношения чрез UML диаграми.

2/Дизайн:

- Клиентски ориентиран дизайн: специфициране на видими за клиента компоненти.
- Имплементационен дизайн: дефиниране на вътрешни алгоритми и структура.

3/Имплементация и интеграция:

- Кодиране и тестване на индивидуални компоненти.
- Интегриране на компонентите в работеща система.

4/Пакетиране: Обединяване на софтуера и документацията в завършен продукт.

Непрекъснати дейности в разработката на софтуер:

-Анализ на риска: Идентифициране на потенциални проблеми.

-Планиране: Определяне на цели и разпределение на ресурси.

-Верфикация: дали продуктът съответства на зададените спецификации (черна кутия и бяла кутия).

-Валидация: дали продуктът отговаря на нуждите на клиента

-Документация

Методологии за разработка:

- Forward engineering- процес на преминаване от високо ниво на абстракция и логически, независими от имплементацията дизайни към физическата имплементация на система.

- **Reengineering-** Изследване на същест. система с цел да се преобразува в нова форма и последващата имплемент. новата форма

-**Agile:** кратки итерации, гъвкавост и бърза адаптация към промени.

-**Scrum:** метод за управл.на проекти с **фокус върху екипната работа**

-**Rational Unified Process (RUP 4+1):** акцент върху архитектурата и use cases; Итеративен и инкрементален; Milestones, Phases, Releases; Inception/старт/ Elaboration/ развитие/ Construction Transition/ преход/ на всяка фаза; A Risk-Driven Approach

-**Reverse Engineering / IBM Rose™/:** анализ на съществуващ код за идентифициране на компоненти и създаване на нови модели.

-**Round-Trip Engineering:** синхронизация между код и дизайн диаграми; постепенно разработване софтуер, започвайки или от нов дизайн, или от съществуващ код

Генерализация: извличане на общи x-ки в родителски класове.

Софтуерната разработка е сложен процес, който изисква структурираност, комуникация и итеративен подход. Основният фокус е върху качеството, гъвкавостта и удовлетворяването на клиентските изисквания чрез съвременни методологии и инструменти.