

1.Професионално софтуерно разработване. Етика в софтуерното инженерство(СИ). Teaching the Next Generation Software Engineering

Резюме

Разходите за софтуер на един персонален компютър често са по-големи от разходите за хардуер; Поддръжката на софтуера струва повече, отколкото неговото разработване; Софтуерното инженерство се занимава с cost-effective software development;

Софтуерни продукти:

Generic: Самостоятелни системи, които се предлагат и продават на всеки клиент, който желае да ги закупи; Спецификация-принадлежи на разработчика и решенията за промени в софтуера взема той.

Customized/Персонализирани/: Софтуер, поръчан от клиент, за да отговори на неговите нужди; Спецификацията принадлежи на клиента и той взема решения за необходимите промени в софтуера

Въпрос	Отговор
Какво е софтуер?	Компют. програми, данни и свързана документация.
Какви са характеристиките на добрия софтуер?	Добрият софтуер трябва да предоставя необходимата функционалност и производителност на потребителя и трябва да бъде поддържан, надежден и лесен за използване .
Какво е СИ?	СИ е инженерна дисциплина, която се занимава с всички аспекти на производството на софтуер .
Какви са основните дейности в СИ?	Спецификация, разработка, валидация и еволюция на софтуера.

Въпрос	Отговор
Каква е разликата между СИ и компютърни науки?	Компютърните науки се фокусират върху теорията и основите ; СИ се занимава с практическите аспекти на разработването и доставянето на софтуер.
Каква е разликата между СИ и системно инженерство?	Системното инженерство се занимава с всички аспекти на разработването на компютърни системи, вкл. хардуер, софтуер и процесно инженерство. СИ е част от този по-общ процес.
Какви са основните предизвикателства пред СИ?	Справянето с нарастващото разнообразие, изискванията за намаляване на времето за доставка и разработването на надежден софтуер.
Какви са разходите за СИ?	Около 60% от разходите за софтуер са за разработка , а 40% са за тестване . За персонал. софтуер разходите за еволюция често надвишават за разработка.
Какви промени е направила мрежата в СИ?	Мрежата е довела до софтуерни услуги и възможността за разработване на силно разпределени системи, основани на услуги , напредък в програмните езици и reuse
Характеристика на продукта	Описание
Поддръжка	Софтуерът да се развива и да отговаря на променящите се нужди на клиентите
Надеждност и сигурност	Надеждността е надеждност, сигурност и безопасност. Софтуерът не трябва да причинява физически или икономически щети в случай на неуспех на С. Злонамерени потребители да нямат

Въпрос	Отговор
	възможност да получат достъп до системата или да я повредят.
Ефективност	Софтуерът да използва излишно на системните ресурси като памет и процесорни цикли. Ефективността включва отзивчивост, време за обработка, използване на паметта и др.
Приемливост	Софтуерът трябва да бъде приемлив за типа потребители, за които е предназначен- разбираем, лесен за използване и съвместим с другите С, които те използват.

Общи въпроси, които засягат повечето софтуер

- Хетерогенност** - С работят като **разпределени системи през мрежи**, които включват разл. типове компютри и мобилни устр.
- Бизнес и социални промени**- променя се съществ. софтуер и се разработва бързо нов софтуер
- Сигурност и доверие**- да можем да се доверим на софтуера

Типове приложения:

- Самостоятелни приложения**- работят на локален компютър-вкл. цялата необх. функционалност и не се нуждаят от свързване с мрежа
- Интерактивни приложения, базирани на транзакции**- изпълняват се на отдалечен компютър и се достъпват от потребители от техните компютри или терминали-уеб приложения; приложения за електронна търговия
- Вградени контролни системи**- софтуерни контролни системи, които контролират и управляват хардуерни устройства. Има повече вградени системи, отколкото всякакъв друг тип системи.

Типове приложения:

-Системи за пакетна обработка-бизнес системи, предназначени да обработват данни в големи пакети. Те обработват голям брой индивидуални входни данни, за да създадат съответни изходи.

Системи за развлекателни цели- лична употреба; забавляват потр

Системи за моделиране и симулация- разработват се от учени и инженери за моделиране на физически процеси или ситуации

Системи за събиране на данни-събират данни от околната среда чрез сензори и изпращат тези данни към др. системи за обработка.

Системи от системи-съставени от други софт. системи

Въпроси за професионална отговорност: Конфиденциалност; Компетентност; Права на интелектуална собственост; Злоупотреба

Етичен кодекс на ACM/IEEE -Професионалните общества в САЩ са работили заедно, за да създадат кодекс за етично поведение.

/Association for Computing Machinery (ACM); Institute of Electrical and Electronics Engineers(IEEE) / - **8 принципа:**

ОБЩЕСТВО(PUBLIC) - Софтуерните инженери трябва да действат в обществения интерес.

КЛИЕНТ И РАБОТОДАТЕЛ(CLIENT AND EMPLOYER) - в интерес на клиент и работодател, съобразен с обществения интерес.

ПРОДУКТ(PRODUCT) - продуктите да отговарят на най-високите професионални стандарти

ПРЕЦЕНКА(JUDGMENT) - интегритет и независимост в професионалната си преценка

УПРАВЛЕНИЕ(MANAGEMENT) - Мениджърите в СИ трябва да се придържат към и насърчават етичен подход към управлението на разработката и поддръжката на софтуер.

ПРОФЕСИЯ(PROFESSION) - да насърчават интегритета и репутацията на професията в съответствие с обществения интерес.

КОЛЕГИ(COLLEAGUES) – сптаведливост и подкрепа на колегите

СЕБЕ СИ(SELF) - доживотно учене

Teaching the Next Generation Software Engineering, Alistair Cockburn

СИ може да бъде релевантно и преподавано чрез няколко подхода:

1. **Craft:** СИ е не само наука, но и **занаят**, който изисква опит и умения; преподаването на СИ се фокусира върху развиването на **практическите умения чрез проекти и упражнения**.
2. **Lean Manufacturing:** Принципите на **Lean** производството могат да бъдат приложени в **СИ**, като се стремим към **оптимизация на процесите и минимизиране на разхищенията**. Това включва техники като **непрекъснато усъвършенстване и ефективно използване на ресурси**
3. **Knowledge Acquisition:** - придобиване нови знания и умения
4. **Cooperative Game:** техники за **сътрудничество и работа в екип чрез игрови подходи**. СИ изисква екипна работа и игрите и симулации помагат на студентите да развият умения за работа в екип и за решаване на проблеми съвместно.

Craft = Доживотно задълбочаване на умения в дадена област;

Craft = Lifelong deepening skills in a medium

7 основни crafts в разработката на софтуер:

1. Решаване какво да се изгради
2. Управление (хора и проекти)

3. Моделиране
4. Проектиране на външния вид
5. Проектиране в голям мащаб (архитектиране)
6. Проектиране в малък мащаб (програмиране)
7. Валидиране на работата

Хората учат умения в 3 етапа:

- **Shu Learn** Учене на една техника
- **Ha Collect** Събиране на техники
- **Ri Invent** Изобретяване/Комбиниране на техники

1: The Craft view brings out lifelong learning

2: The Cooperative Game brings out people issues

SE is a Cooperative Game of Invention & Communication ... making ideas concrete in an economic context ...

Хора, които изобретяват и комуникират, решават проблем, който още не разбират, и който продължава да се променя

Създават решение, което още не разбират, и което също продължава да се променя. Изразяват идеи на езици, които не разбират напълно, а тези езици също се променят. Това се случва пред интерпретатор, който не прощава грешки. Въз основа на това те вземат решения с ограничени ресурси, като всяко тяхно решение има икономически последиствия.

Безкрайност: Организационно оцеляване, управление на кариерата

Кооперативната игра учи на екипна работа и стратегии в ситуации, които никога не се повтарят напълно.

Конкурентни и кооперативни игри:

-Ограничени без фиксиран край: Джаз музика, борба за върха (King-of-the-hill)- играчите не знаят точно кога ще приключи.

-Ограничени и целенасочени: Тенис, Покер, Шах, Театър, Инженерство, Алпинизъм

Разработката на софтуер е серия от игри. Конфликтни цели се конкурират за ресурси: 1/ доставяне система 2/ Настройване за следващата игра Няма проста формула за победа, а само стратегии за конкретни ситуации! (Shu-Ha-Ri)

3: Lean производството може да научи дизайнерите на много.
Design=manufacturing if inventory=unvalidated decisions

-Мрежата от хора, чакащи другите, има цикли.

-Различни стратегии произтичат от местоположението на опашката.

-Lean процесите използват по-малки стъпки и партии.

4: Knowledge Acquisition - СИ вкл. стратегич. придобиване знания
Дизайнът е игра на стратегическо обучение

Платете, за да научите рано

Разработвайте за бизнес стойност по пътя.

Trim the Tail: Доставка по стойност или по дата.

Неуспех на СИ в съвременността; Липса на ефективност на традиционния модел(1968 г.) на софтуерното инженерство,
Необходимост от нов модел