

5. Управление на качеството(УК)-Резюме

УК- гарантиране, че необходимото ниво на качество(К) е постигнато в софтуерен продукт. 3 основни аспекта:
-**На организационно ниво**- установяването на рамка от **организационни процеси и стандарти**, които ще доведат до **висококачествен софтуер**.

-**На проектно ниво**- УК вкл. прилагането на специф. п-си за **осигуряване на К** и проверка дали планираните п-си са били **спазени**. Изготвянето на **план за К на проекта**- да определи целите за **К** и да дефинира какви **процеси и стандарти** ще бъдат използвани.

Дейности по управление на качеството:

- УК осигурява независима **проверка на п-са** на разработка на софтуер
- Процесът на УК **проверява резултатите от проекта**, за да гарантира, че те са в **съответствие с организационните стандарти и цели**.
- Екипът по К трябва да бъде **независим от екипа по разработка**, за да може да има обективен поглед. Това позволява да докладват за К на софтуера, без да бъдат повлияни от въпроси, свързани с разработката.

Планиране на качеството

- Планът за качество(ПК)** определя **желаните х-ки на продукта**, начина, по който те се оценяват, и най-значимите качеств.атрибути.
- ПК трябва да дефинира процеса на оценка на К.
- ПК трябва да посочи кои организационни стандарти трябва да се прилагат и да определи стандарти за използване.

Структура на ПК: Представяне продукта; Планове за продукта; Описания на процесите; Цели за К; Риск и управление на риска

ПК да са **кратки и ясни** документи

Обхват на УК-УК е особено **важно** за големи и сложни С. Документацията за К е запис на напредъка и подпомага непрекъснатостта на разработката, дори когато екипът се променя. За по-малки С УК изисква **по-малко документация** и трябва да се фокусира върху изграждането на **култура на К**.

Качество на софтуера- даден **продукт** трябва да отговаря на своята **спецификация**. Напрежение м/у И за К на клиента (ефективност, надеждност) и И за К на разработчиците (поддръжка, reuse);

- И за К е трудно да се специфицират по недвусмислен начин;
- **Софтуерните спецификации са непълни и често несъгласувани**

Фокусът може да бъде върху "**fitness for purpose**", а не само върху **съответствие със спецификациите**.

Пригодност на софтуера за употреба - fitness for purpose

-Спазени ли са **стандартите за програмиране и документация** в процеса на разработка? Софтуерът **тестван ли е правилно**? Достатъчно **надежден ли е софтуерът**, за да бъде внедрен? Софтуерът **удобен ли е** за използване? Софтуерът **добре структуриран и разбираем ли е**?

Атрибути на софтуерното качество (Software quality attributes)

Safety – Безопасност; **Understandability** – Разбираемост

Portability – Преносимост; **Security** – Сигурност

Testability – Тестируемост; **Usability** – Удобство за използване

Reliability – Надеждност; **Adaptability** – Адаптивност

Reusability – Повторна използваемост; **Resilience** – Устойчивост;

Modularity – Модулност; **Efficiency** – Ефективност

Robustness – Робустност (Здравина); **Complexity** – Сложност

Learnability – Леснота на научаване

Конфликти на качеството-Невъзможно е дадена С да бъде оптимизирана за всички тези атрибути –подобряването на робустността може да доведе до загуба на производителност.

-ПК да определи **най-важните качествени атрибути** на софтуера

-ПК да включва **дефиниция на процеса за оценка на К и договореност за оценка кач. х-ки** /pr.maintainability or robustness/

Качество на процеса и продукта-К на разработения продукт се влияе от К на производствения п-с.

- някои **атрибути на К на продукта** са трудни за оценка.

-връзката м/у софт. п-си и К на продукта е сложна и слабо разбрана

-**Уменията и опитът на разработчиците** играят ключова роля

-**Външни фактори**, като **новост на дад. приложение** или **необходимост от ускорен график** за разработка, могат да влошат КП

Софтуерни стандарти-определят **изискваните атрибути на даден продукт или процес**. Те играят важна роля в УП. Стандартите могат да бъдат **международни, национални, организационни или проектни**.

-**Продуктовите** стандарти определят х-ките, които всички софт. компоненти трябва да притежават, пр. **общ стил на програмиране**.

-**Процесните** стандарти дефинират как да се изпълнява **софт. п-с**.

Значение на стандартите -Обобщаване на добрите практики –да не се повтарят минал.грешки-**рамка за дефиниране на К** – виждането

на организацията за К. **Осигуряват приемственост** – новите служители могат по-лесно да разберат организацията чрез стандартите, които тя използва.

Проблеми със стандартите

- Софт. инженери** може да не ги приемат за **актуални и подходящи**.
- Често включват **твърде много бюрократични формалности**.
- Ако не са **подкрепени от софтуерни инструменти**, поддържането на **свързаната документация** става досадна и времеемка задача.

Разработване на стандарти

- Включване на **практикуващи специалисти** в разработването – инженерите трябва да разбират **логиката зад стандарта**.
- Редовен преглед** на стандартите и тяхн. приложение– те бързо могат да **остарееят**-намалява доверието на специалистите към тях.
- Подробните стандарти трябва да бъдат **подкрепени със специализирани инструменти**. Най-честото оплакване срещу стандартите е **прекомерната административна работа**.

Рамка на стандартите ISO 9001 - Международен набор от стандарти, използвани като **основа за разработв. на системи за УК**.

ISO 9001, най-общият от тях, се прилага за организации, които **проектират, разработват и поддържат продукти вкл. софтуер**

Стандартът **ISO 9001** -рамка за разработване на **софт. стандарти**:

- Определя общи **принципи** за качество;
- Описва **качествени процеси** като цяло;

- Задава **организационни стандарти и процедури**, които трябва да бъдат дефинирани и документирани в **организационно ръководство за качество**.

Сертификация по ISO 9001

-Стандартите за качество и процедурите трябва да бъдат документирани в **организационно ръководство за качество**.

-Външна организация може да **сертифицира**, че **ръководството за К** на дадена организация отговаря на стандартите **ISO 9001**.

-Някои клиенти **изискват** от своите доставчици да бъдат сертифицирани по **ISO 9001**, въпреки че необходимостта от **гъвкавост** в това отношение се признава все повече.

Ключови моменти

-**Управлението на софтуерното качество (УСК)** има за цел да гарантира, че **софтуерът** има **малък брой дефекти** и че отговаря на изискванията за **поддръжка, надеждност, преносимост** и др.

УСК включва: Дефиниране на **стандарти за процеси и продукти**;
Създаване на п-си за проверка на спазването на тези стандарти.

Софтуерните стандарти са от същ. значение за **осигуряване на К**, тъй като представляват **най-добри практики**.

Процедурите за УК могат да бъдат документирани в **организационно ръководство за качество**, базирано на **общия модел за ръководство за качество** в стандарта **ISO 9001**.

Прегледи и инспекции: Група преглежда част или целия процес на С и нейната документация, за да открие потенциални проблеми. Софтуерът или документите могат да бъдат "одобрени- signed off " на преглед, което означава, че напредъкът към следващия етап на

разработка е одобрен от ръководството. Съществуват различни видове прегледи с различни цели:

§ Инспекции за отстраняване на дефекти (продукт);

§ Прегледи за оценка на напредъка (продукт и процес);

§ Прегледи за качество (продукт и стандарти).

Прегледи за качество-Група хора внимателно преглеждат част или целия софтуер и свързаната с него документация. Код, дизайни, спецификации, тестови планове, стандарти и други могат да бъдат прегледани. Софтуерът или документите могат да бъдат "одобрени" на преглед, което означава, че напредъкът към следващия етап на разработка е одобрен от ръководството.

Прегледи и гъвкави методи: Процесът на прегледи в гъвкавото софтуерно развитие обикновено е неформален:

-В Scrum има среща за преглед след всяка итерация на софтуера (преглед на спринта), където могат да бъдат обсъдени проблеми с К.

-В XP pair-програмирането гарантира, че кодът постоянно се преглежда от друг член на екипа. XP разчита на индивидуалната инициатива за подобряване и рефакториране на кода. Гъвкавите подходи обикновено не се ръководят от стандарти, така че въпросите за съответствието със стандартите обикновено не се вземат предвид.

Инспекции на програми-прегледи м/у колеги, при които инженери разглеждат кода на система с цел откриване на аномалии и дефекти.

-не трябва изпълнение на С и могат да се използват преди реализация

-могат да се прилагат към всяко представяне на системата (изисквания, дизайн, конфигурационни данни, тестови данни и т.н.).

- ефективна техника за откриване на грешки в програмите

Контролен списък за инспекция с често срещани грешки трябва да се използва за насочване на **инспекцията**. Контролни списъци за грешки **зависят от езика** за програмиране и отразяват **характерн. грешки**, които е вероятно да възникнат в него. По-"слабо" типизиране в езика- по-голям списък за проверка

Гъвкави методи и инспекции - Гъвкавите процеси рядко използват формални инспекции или проверки между колеги.

- гъвкавите методи разчитат на сътрудничеството между членовете на екипа, които проверяват кода си взаимно, както и на неформални насоки-„провери преди качване“, което насърчава програмистите да проверяват сами кода си.

-Практикуващите ХР твърдят, че програмирането по двойки е ефективен заместител на инспекцията, тъй като по същество представлява непрекъснат процес на проверка. Двама души разглеждат всяка линия код и я проверяват, преди е приета.

Измерване и метрики на софтуера - извличането на числова стойност за даден атрибут на софтуерен продукт или процес. Позволява обективни сравнения между разл. техники и процеси. Някои компании са въвели програми за измерване. Повечето организации все още не прилагат систематично измерване на софт. В тази област има малко установени стандарти.

Метрика на софтуера- Всеки тип измерване, който се отнася до софтуерна система, процес или свързана документация: Брой редове код в програма, индекс на Fog, брой човекодни за разработването на даден компонент.

- Позволяват количествено измерване на софтуера и софт. процес.
- Използват за прогнозиране на атрибути на продукта или за контрол на софт. процес.

- Метриките на продукта могат да се използват за общи прогнози или за идентифициране на аномални компоненти.

Използване на измервания- Присвояване на стойност на атрибутите за К на С - Чрез измерване на х-ките на системните компоненти, като тяхната цикломатична сложност/ сложност на кода/, и събиране тези измервания, може да се оценят атрибути на качеството на системата, като напр. поддръжка.

-Идентифициране на компоненти в С с ниско К - Измерванията могат да идентифицират отделни компоненти, чиито характеристики се отклоняват от нормата. Може да се измерят компонентите, за да се открият тези с най-висока сложност. Те най-вероятно съдържат грешки, защото високата сложност ги прави по-трудни за разбиране.

Предположения при метриките

-Свойствата на софтуера могат да бъдат измерени.

- Съществува връзка между това, което можем да измерим, и това, което искаме да знаем. Можем да измерваме само вътрешни атрибути, но често сме заинтересовани от външн. атрибути на софт.

- Тази връзка е формализирана и валидирана.

- Възможно е да бъде трудно да се свърже това, което може да бъде измерено, с желаните външни атрибути на качеството.

Проблеми с измерването в индустрията

- Невъзможно е количествено да се определи възвръщаемостта на инвестицията от въвеждането на организационна програма за метрики.

- Липсват стандарти за софтуерни метрики или стандартизирани процеси за измерване и анализ.

- В много компании софтуерните процеси не са стандартизирани и са слабо дефинирани и контролирани.

- Повечето изследвания върху измерването на софтуера са фокусирани върху метрики, базирани на кода, и планово управлявани процеси. Въпреки това все повече софтуер се разработва чрез конфигуриране на ERP системи или COTS.

- Въвежд. измервания добавя допълнителни разходи към процесите.

Метрики на продукта -метриките за К трябва да предсказват К на продукта. Класове метрики на продукта:

§ **Динамични** метрики- измервания на **изпълняваща** се програма;

§ **Статични** метрики- измервания на **представянето** на системата;

§ Динамичните метрики помагат за оценка на ефективността и надеждността;

§ Статичните метрики помагат за оценка на сложността, разбираемостта и поддръжката.

Динамични и статични метрики

- **Динамичните** метрики са тясно свързани с **атрибутите на качеството на софтуера**- измерване времето за отговор на С (атрибут на производителност) или броят на отказите (атрибут за надеждност)

- **Статичните** метрики имат индиректна връзка с атрибутите на К- Трябва да се опитае да извлечем връзка между тези метрики и свойства като сложност, разбираемост и поддръжка.

Статични метрики на софтуерния продукт: Fan-in/Fan-out – мярка за броя на функциите или методите, които извикват друга функция или метод; **Дължина на кода; Цикломатична сложност; Дължина на идентификаторите; Дълбочина на условното**

влагане; Fog индекс – **мярка за средната дължина на думите и изреченията в документи.**

ОО- метрики на УК: Тежки методи на клас; Дълбочина на дървото на наследяване; Брой наследници; Свързаност между обектни класове; Отговор на клас; Липса на сплотеност в методите

Анализ на софтуерни компоненти - Системните компоненти могат да бъдат анализирани поотделно, използвайки различни метрики. Стойностите на тези метрики могат да се сравняват за разл. компоненти и с исторически измерени данни от предишни проекти. Аномални измервания, които значително се отклоняват от нормата, могат да означават проблеми с качеството на тези компоненти.

Неочаквани резултати от измерванията-Намаляването на броя на грешките в програмата води до увеличаване на броя на обажданията към помощния център: Програмата вече се възприема като по-надеждна и достига до по-широк и разнообразен пазар.

Ключови моменти

- **Прегледите** на резултатите от софтуерния процес включват екип от хора, които проверяват дали стандартите за качество се спазват.
- При **инспекция на програма** или **преглед между колеги** малък екип систематично проверява кода, като го чете в детайли и търси възможни грешки и пропуски.
- **Измерването на софтуера** може да се използва за събиране на данни за софтуера и софтуерните процеси.
- **Метриките за К на продукта** са особено полезни за открояване на **аномални компоненти**, които могат да имат проблеми с качеството.