

Модели за качество на софтуера

Доц. д-р Ася Стоянова-Дойчева



Съдържание

- Общи понятия
- Модел на качеството на софтуера
- Типичен йерархичен модел
- Определяне стойностите на оценъчните елементи
- Процедура за оценяване
- Предимства и недостатъци на йерархичния модел



Общи понятия

- “Software Engineering е дисциплина, която предоставя методи и средства за разработване на **качествен софтуер** с ограничен бюджет, поставени срокове и с постоянно изменящи се изисквания. “

IEEE

- Какво е качество???



Общи понятия

“Качеството е съвкупност от средства и характеристики на даден продукт или услуга, носители на способността му да отговори на явно или неявно указани нужди.”

ISO 8402-1986



Модел на качеството на софтуера

- Модел на Боем – 1973г. – 1978 (доусъвършенстван).
- Моделът има йерархичен характер с две нива – полезност и възможност за лесно съпровождане
- Полезността се определя от следните характеристики:
 - Надеждност
 - Ефективност
 - Използваемост
- Възможността за лесно съпровождане се определя от характеристиките:
 - Тестируемост
 - Разбираемост
 - Модифицируемост
- Портатилността



Модел на качеството на софтуера

- Надеждността се определя от свойствата:
 - Пълнота
 - Точност
 - Непротиворечивост
- Разбираемостта също се определя от свойството
 - Непротиворечивост



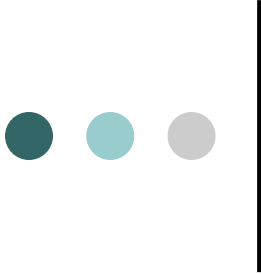
Модел на качеството на софтуера

- Всяко свойство за конкретния програмен продукт да се оцени с някаква обективна мярка. Тази мярка се нарича **метрика**.
 - Важно е да се определи важността на метриката.
 - Важно е да се автоматизира процеса на оценяване на свойствата и характеристиките.



Модел на качеството на софтуера

- Недостатъци на модела на Боем:
 - Не съвсем ясна структурираност
 - Недостатъчна пълнота на множеството от характеристики
 - Съсредоточаване почти изцяло върху качеството на кода, а не върху целия продукт
 - Сравнително тясна експериментална база (програми на Фортран)



Типичен йерархичен модел

- **Качеството** се разглежда като йерархична структура. То се намира на ниво 0
- На следващо ниво 1 се намират **факторите**, които са потребителско ориентирано свойство, представящо даден аспект на качеството на софтуера от гледна точка на потребителя
- На ниво 2 се намират **критериите** – те са софтуерно ориентирани свойства, представящи характеристики на програмния продукт.
- На 3 ниво се намират **метриките** – те са софтуерно ориентирани детайли на даден критерии. Те се определят от оценъчните елементи.
- Последно ниво 4 е на **оценъчните елементи** – те са елементарни характеристики на най-ниско ниво, които подлежат на количествена оценка.



Фактори

- В зависимост от модела те могат да бъдат от 6 до 16. В нашия случай са 6:
 - Гъвкавост
 - Коректност
 - Надеждност
 - Съпровождаемост
 - Удобство на използване
 - Ефективност



Критерии

- Всеки фактор се определя от няколко критерия. Например фактора съпровождаемост (възможност за отстраняване на грешки и поддръжката му в актуално състояние) се определя от критериите:
 - Структурираност
 - Простота
 - Нагледност



Метрики

- Те се определят от оценъчните елементи. Нека разгледаме като пример метриките за критерия нагледност:
 - Коментари към логиката на програмата
 - Оформяне на текста на програмата
 - Възприета система за идентификация



Оценъчни елементи

- Те са елементарна характеристика на най-ниско ниво, която подлежи на количествено оценяване. Нека разгледаме метриката коментари към логиката на програмата за критерия нагледност. Тя се определя от следните оценъчни елементи:
 - Коментари към машиннонезависимите елементи на програмата;
 - Коментари към машиннозависимите елементи на програмата;
 - Коментари към входно-изходните точки.

Йерархична структура на модела за качеството

Ниво 0

Качество

Ниво 1

Гъвкавост

Коректност

Надеждност

Съпровожда
емост

Удобство на
ползване

Ефективност

Ниво 2

Структурира
ност

Простота

Нагледност

Ниво 3

Оформление
на текста

Коментар на
логиката на
програмите

Система за
идентификац
ия

Ниво 4

Коментар
към

Коментар
към

Коментар
към.....



Определяне стойностите на оценъчните елементи

- Методите за определяне стойностите на оценъчните елементи могат да се класифицират по два признака:
 - По начина на получаване на информацията за програмния продукт
 - По източника на получаване на информацията



Определяне стойностите на оценъчните елементи

- По начина на получаване на информацията за продукта са:
 - **Измерителен** – състои се в използването на програмни инструменти за определяне обема на програмата, на времето на изпълнението на цялата програма или определени класове, на времето на реакция на програмата на определени входове и др.
 - **Регистрационен** – основава се на информация получена по време на изпитания или експлоатация на програмата
 - **Органолептичен** – основан е на използването на информация получена от човека в резултат на анализа на възприятията му.
 - **Изчислителен** – използва теоретични и емпирични зависимости, статистически данни и др.



Определяне стойностите на оценъчните елементи

- По източника на получаване на информацията методите са:
 - Традиционен – стойностите се определят от специализирани организации за изпитания и изчисления.
 - Експертен – стойностите се определят от група експерти компетентни в решаването на определен вид задача.
 - Социологически – стойностите се получават чрез разработването на специални анкети въпросници.



Процедура за оценяване

1. Всяка характеристика може да приема стойност между $[0,1]$.
 - 1.1 Стойностите на всички оценъчни елементи се определят от експерти по един от споменатите вече методи.
 - 1.2 Дават се указания на експертите – ако се отнася за получаване на стойност по измерителен, регистрационен или изчислителен метод се посочва точният начин за получаване на стойността. В останалите случаи се дават по-обща указания. Ако вземем за пример коментарите към входно-изходните точки:
 - 0, ако изцяло липсват;
 - 0.33, ако ги има, но са твърде кратки и неясни
 - 0.67, ако ги има и са задоволителни
 - 1, ако са отлични



Процедура за оценяване

2. На всяка характеристика от всяко ниво съответства тегло в интервала $[0,1]$. Сумата от теглата на характеристиките, определящи характеристика от по-високо ниво е 1. Нека разгледаме теглата на ниво оценъчни елементи за метриката коментари към логиката на програмата:
- 0.4 за коментари към машиннонезависимите елементи на програмата;
 - 0.3 за коментари към машиннозависимите елементи на програмата;
 - 0.3 за коментари към входно-изходните точки.
- 2.1 Всички теглови стойности се определят от експерти и се отнасят за точно определен тип софтуерни продукти.



Процедура за оценяване

3. При започване на оценката на качеството на софтуера ще разполагаме с :

- Процедура за оценяване
- Указания за намиране на стойностите на оценъчните елементи
- Теглата към всички характеристики на всички нива за дадения тип софтуер

4. Експертите определят стойностите на оценъчните елементи.



Процедура за оценяване

5. Нека получените стойности на оценъчните елементи определящи метрика M са e_1, e_2, \dots, e_n , а съответните им предварително зададени тегла са w_1, w_2, \dots, w_n . \Rightarrow

$$M = e_1 * w_1 + e_2 * w_2 + \dots + e_n * w_n$$

Същата схема на пресмятане се прилага и за по-горните нива.



Процедура за оценяване

6. След като са известни всички стойности на метриците M за всеки критерии C прилагаме формулата:

$$C = M_1 * w_1 + M_2 * w_2 + \dots + M_n * w_n,$$

където M_i са метриците, определящи критерия C .



Процедура за оценяване

7. Аналогично след като всички стойности на критериите C са известни, за да получим стойността на всеки фактор F прилагаме формулата:

$$F = C_1 * w_1 + C_2 * w_2 + \dots + C_n * w_n ,$$

където C_i са критериите, определящи фактора F .



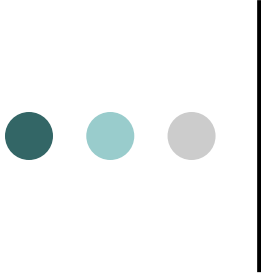
Процедура за оценяване

8. Накрая, когато всички стойности на фактори F са известни, за да получим стойността на качеството Q , прилагаме формулата:

$$Q = F_1 * w_1 + F_2 * w_2 + \dots + F_n * w_n,$$

където F_i са факторите, определящи качеството Q .

Q остава в интервала $[0,1]$.



Предимства и недостатъци на йерархичния модел

- Предимства
 - Простота
 - Конструктивност
 - Автоматизация
 - Крайният резултат е едно число в интервал
- Недостатъци
 - Субективност
 - Трудоемкост
 - Всеки тип програмен продукт изисква свое собствено множество от тегла