



Тема 10

Измерване и метрики



Съдържание

- Основи на теорията на измерването
 - Абстрактни концепции, дефиниции, измервания
 - Нива на измерване
- Метрики



ОСНОВНИ ПОНЯТИЯ

■ Измерване

- Процес, при който в съответствие с определени правила, на характеристики на изследвания обект се съпоставят стойности.

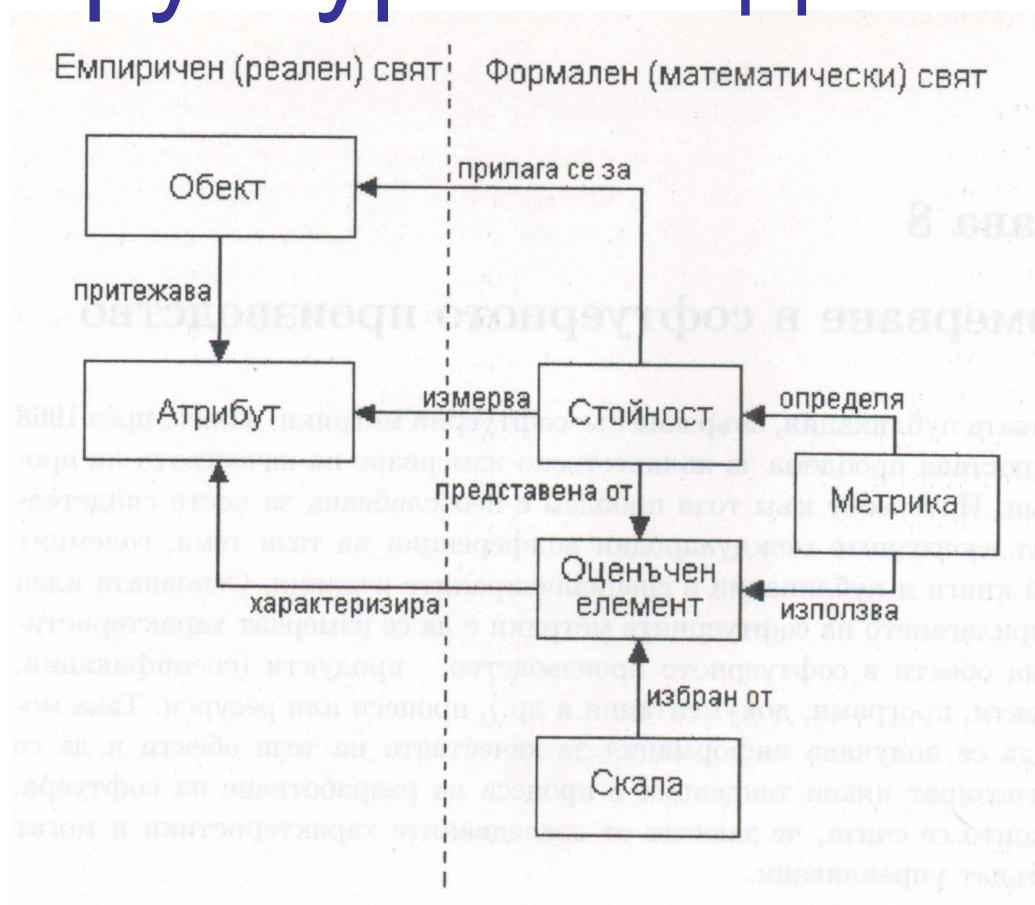
■ Мярка

- Числото, съпоставено при измерването
- Мярката представя различните състояния на измерваната характеристика

■ Метрика

- = мярка
- Процедура за намиране и използване на метрика

Софтуерно измерване структурен модел





Нива на измерване

- Номинална (Nominal) скала
- Порядкова (Ordinal) скала
- Интервална (Interval) скала
- Относителна (Ratio) скала



Номинална (Nominal) скала

- Изброимо множество от категории/стойности, които се присвояват, без да се вземат предвид количествени съотношения
- Категориите не могат да се използват във формули
- 2 изисквания към категориите
 - Взаимно изключващи (mutually exclusive)
 - Взаимно изчерпателни (jointly exhaustive)
- Не могат да се превръщат мерки от една към друга номинална скала



Порядкова (Ordinal) скала

- Наредено множество от категории, позволяващо подреждане на стойностите на оценъчните елементи чрез сравняването им с определени гранични стойности.
- Порядковата скала е асиметрична
 - If $A > B$ is true then $B > A$ is false
- Порядковата скала има транзитивно свойство
 - If $A > B$ and $B > C$, then $A > C$
- Нямаме информация за степента на разлика между елементите



Интервална (Interval) скала

- Числови стойности, като разликата между всеки две последователни е една и съща
- Могат да се прилагат математически операции (събиране и изваждане) върху данните от скалата
- Могат да се превръщат мерки за един и същи атрибут от една към друга интервална скала



Относителна (Ratio) скала

- Както интервалната, но освен че разликата между две последователни стойности е една и съща, има и ясно дефинирана мярка 0.
- Всички математически операции могат да се прилагат върху данните от скалата



Свойства на измерването

- **Обективност**
 - Получаваните мерки да не зависят от субекта, извършващ измерването
- **Надеждност**
 - При повтаряне на измерването при еднакви условия да се получават еднакви резултати
- **Валидност**
 - Получаваните мерки да отразяват реално свойствата на измервания обект
- **Точност**
 - Мярката да има необходимата различаваща способност



Софтуерно измерване

- Дейности
 - Формулиране на метричната система
 - Събиране на данни
 - Анализирание
 - Интерпретиране
 - Осъществяване на обратна връзка



Съдържание

- Основи на теорията на измерването
 - Абстрактни концепции, дефиниции, измервания
 - Нива на измерване
- Метрики



Дефиниции на метрики за качеството

- IEEE, 1990
 - Качествена мярка на степента до която определен артикул притежава даден атрибут на качеството
 - Функция, чиито входове са софтуерни данни и чиито изход е една числова стойност, която може да се интерпретира като степента, до която даден атрибут влияе върху качеството на софтуерните процеси или продукти



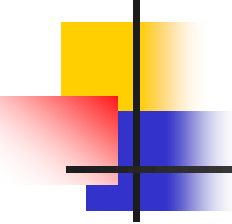
Общи изисквания

- Уместна /relevant/
 - Измерва атрибут от голямо значение
- Валидна /valid/
 - Измерва искания атрибут
- Надеждна /reliable/
 - Дава еднакви резултати, когато се прилага в еднакви условия
- Изчерпателна /comprehensive/
 - Приложима в различни ситуации
- Взаимно изключваща /mutually exclusive/
 - Не измерва атрибути, вече измервани от други метрики



Оперативни изисквания

- Лесно и просто
- Не изисква независимо събиране на данни
 - Базира се на текущо използваните системи за събиране на данни
- Неподатлива на предубедени намеси от заинтересовани страни



Класификация на софтуерните метрики

- В зависимост от предназначението
- В зависимост от целта на прилагане на метриката
- В зависимост от типа на изследвания обект
- В зависимост от начина на получаване на информацията



Класификация в зависимост от предназначението

- Оценяване на разходите и усилията за разработване на софтуера
- Измерване на производителността на разработчиците
- Моделиране на качеството на софтуера и оценяването му
- Измерване и прогнозиране на надеждността на създаваните софтуерни системи

Класификация в зависимост от целта

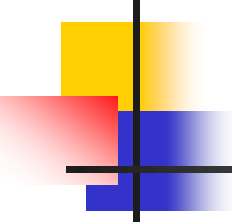


- За характеризиране
 - Да разберем изследвания обект и да установим база за сравнение при бъдещи оценки
- За оценяване (assess)
 - Регистриране на текущо състояние
- За прогнозиране (predict)
 - Предварителна оценка за бъдещо състояние
- За подобряване
 - Натрупване на количествена информация за определяне на факторите, от които зависи качеството

Класификация в зависимост от начина на получаване на информацията



- Регистрационен подход
- Измерителен подход
- Възприятен подход
 - Информация се получава след анализ на зрителни или слухови възприятия
- Изчислителен метод
 - Използване на теоретични и/или емпирични зависимости, изведени въз основа на статистически данни



Класификация в зависимост от типа на обекта - 1

- Метрики на продукта
 - свързани са с характеристики като: размер, сложност, характеристики на дизайна, бързодействие
- Метрики на процеса
 - Служат за оценка на процеса за разработка – на базата на анализ -> отстраняване на несъответствия, повишаване на ефективност
- Метрики на проекта
 - свързани са с въпросите, засягащи частта, включваща дисциплината **Управление на проекти** - финансовите резултати, времевите резултати, продуктивността на екипа

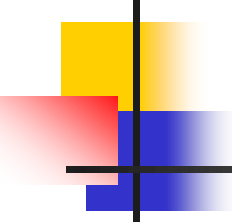
Забележка: Някои метрики могат да се отнесат към повече от една категория.

ОБЕКТИ	ВЪТРЕШНИ ХАРАКТЕРИСТИКИ	ВЪНШНИ ХАРАКТЕРИСТИКИ
ПРОДУКТИ		
Спецификации	Размер Модулност Коректност Функционалност Възможност за повторно използване	Изчерпателност Съпровождаемост Разбираемост
Проекти	Размер Модулност Свързаност	Качество Сложност Съпровождаемост
Програми	Размер Алг. сложност Тестируемост	Надежност Приложимост Съпровождаемост
Тестови данни	Размер Ниво на тестване	Качество Изчерпателност
Документация	Обем Пълнота	Разбираемост Адекватност
ПРОЦЕСИ		
Специфициране	Продължителност Трудоемкост Брой изменения в изискванията	Качество Разходи Стабилност
Детайлно проектиране	Продължителност Трудоемкост Брой открити грешки в спецификациите	Разходи
Тестване	Продължителност Трудоемкост Брой открити грешки в спецификациите	Разходи Изчерпателност Систематичност
РЕСУРСИ		
Кадри	Възраст Възнаграждение	Производителност Професионален опит
Екип	Численост Структура	Продуктивност Качество
Софтуер	Цена Размер	Полезност Надежност
Хардуер	Цена Изч. възможности	Надежност на функциониране



Подкатегории на метрики на процес

- Метрики за **качество** на софтуерен процес
- Метрики за **времеви график** на софтуерен процес
- Метрики за **ефективност** на отстраняване на грешки
- Метрики за **продуктивност** на софтуерен процес



Метрики за качество на софтуерен процес

- Две категории метрики за качество на софтуерен процес
 - Метрики за **честота на грешките** (error density)
 - Размер на софтуера (Software volume)
 - Брой грешки (Errors counted)
 - Метрики за **сериозност на грешките** (error severity)

Метрики за честота на грешките

Code	Name	Calculation formula
CED	Code Error Density	$CED = \frac{NCE}{KLOC}$
DED	Development Error Density	$DED = \frac{NDE}{KLOC}$
WCED	Weighted Code Error Density	$WCED = \frac{WCE}{KLOC}$
WDED	Weighted Development Error Density	$WDED = \frac{WDE}{KLOC}$
WCEF	Weighted Code Errors per Function point	$WCEF = \frac{WCE}{NFP}$
WDEF	Weighted Development Errors per Function point	$WDEF = \frac{WDE}{NFP}$

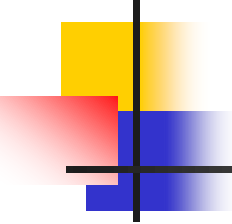
Key:

- NCE = number of code errors detected in the software code by code inspections and testing. Data for this measure are culled from code inspection and testing reports.
- KLOC = thousands of lines of code.
- NDE = total number of development (design and code) errors detected in the software development process. Data for this measure are found in the various design and code reviews and testing reports conducted.
- WCE = weighted code errors detected. The sources of data for this metric are the same as those for NCE.
- WDE = total weighted development (design and code) errors detected in development of the software. The sources of data for this metric are the same as those for NDE.
- NFP = number of function points required for development of the software. Sources for the number of function points are professional surveys of the relevant software.



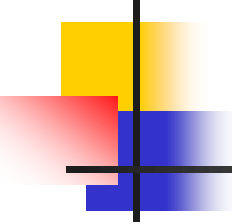
Метрики за сериозност на грешките

- Average severity of code errors
 - $ASCE = WCE / NCE$
- Average severity of development errors
 - $ASDE = WDE / NDE$



Метрики за времеви график на софтуерен процес - 1

- Два алтернативни подхода
 - На базата на отчети за успех (завършени междини резултати по график) в допълнение на събития на неуспех (незавършени в график)
 - На база на средно закъснение при завършване на междинни резултати



Метрики за времеви график на софтуерен процес - 2

- Time table observance
 - $TTO = MSOT / MS$
- Average delay of milestone completion
 - $ADMC = TCDAM / MS$

MSOT – milestones completed on time

MS – total number of milestones

TCDAM – total completion delays for all milestones



Метрики за ефективност на отстраняване на грешки - 1

- Измерват се от система за осигуряване на качество след период от работа на софтуерната система
- Метриките комбинират записите за грешки по време на разработване със записите за повреди събрани по време на определен период на постоянна работа

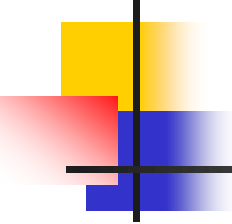


Метрики за ефективност на отстраняване на грешки - 2

- Development Errors Removal Effectiveness
 - $DERE = NDE / (NDE + NYF)$
- Development Weighted Errors Removal Effectiveness
 - $DWERE = WDE / (WDE + WYF)$

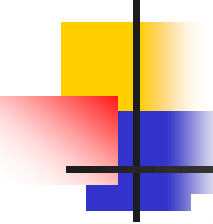
NYF – number of software faults detected during a year of maintenance service

WYF - weighted number of software faults detected during a year of maintenance service



Метрики за продуктивност на софтуерен процес - 1

- Директни
 - Продуктивност на човешките ресурси в проектите
- Индиректни
 - Фокусират се върху степента на повторно използване



Метрики за продуктивност на софтуерен процес - 2

Code	Name	Calculation formula
DevP	Development Productivity	$\text{DevP} = \frac{\text{DevH}}{\text{KLOC}}$
FDevP	Function point Development Productivity	$\text{FDevP} = \frac{\text{DevH}}{\text{NFP}}$
CRe	Code Reuse	$\text{CRe} = \frac{\text{ReKLOC}}{\text{KLOC}}$
DocRe	Documentation Reuse	$\text{DocRe} = \frac{\text{ReDoc}}{\text{NDoc}}$

Key:

- DevH = total working hours invested in the development of the software system.
- ReKLOC = number of thousands of reused lines of code.
- ReDoc = number of reused pages of documentation.
- NDoc = number of pages of documentation.



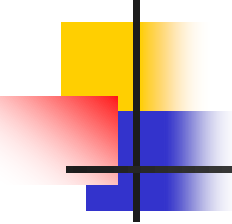
Метрики за продукт

- Средно време до отказ (**MTTF**)
- Процент на грешки (**Defect rate**)
 - определя на базата на допуснатите грешките спрямо възможностите за допускането им (**O**pportunities **F**or **E**rror), изразени например в брой линии от кода (Lines Of Code –LOC), брой функционални точки (**F**unctional **P**oints).
 - FP е набор от изпълними оператори, които изпълняват определена задача, заедно с декларациите на формалните параметри и локалните променливи, използвани в тях.
- Проблеми, идентифицирани от потребителите (**Customer problems**).
 - могат да бъдат действителни грешки, а могат да бъдат и свързани с неясна документация, или резултат от непознаване на системата.
 - най-често се използва Проблеми за месеци, в които потребителят експлоатира системата - Problems per User Month (PUM).
- Удовлетвореност на потребителите (**Customer satisfaction**)
 - анкети на определени периоди след внедряването на софтуерното решение – много удовлетворен, удовлетворен, неутрален ...



Метрики за поддръжка

- Индекс за управление на грешките
 - Брой получени отчети за грешки / брой отстранени за фиксиран период от време
- Време за отстраняване на грешките
- Процент грешки, чието отстраняване е просрочено във времето
- Брой погрешно обработени грешки



Грешки, регистрирани в резултат на:

- Преглед/ревью на кода (Code review);
- Преглед на документацията – в спецификациите за проекта или в потребителската документация;
- Преглед на фазите и инспекция на проекта;
- Тестове на системата по време на разработката ѝ;
- Тестове на системата при приемането ѝ;
- Работа на системата в гаранционен период.



Описание на грешката в проект

- Необходимо да се посочи **къде е открита**:
 - Да се опише името на проекта;
 - Компонент, в който е открита;
 - Номер на компилация (build), в която се регистрира за първи път;
 - Среда: Описание на тестовата платформа: хардуер и софтуер (операционна система и т.н.).
- Състояние на грешката :
 - Нова,
 - Обработка се ...



Описание на грешката в проект 2

- Трябва да се определи **важността на грешката** (Severity), която дава представа
 - какви са последствията от нея
 - Последствията могат да варира от “критични” до “препоръки за подобрене”.
 - определя състоянието на системата, в резултат от нейното наличие.
- Трябва да се посочи **приоритет** (Priority) на грешката, който определя колко бързо тя трябва да се отстрани.
 - Най-често използваните скали за описанието му са: Критичен, Висок, Среден, Нисък.



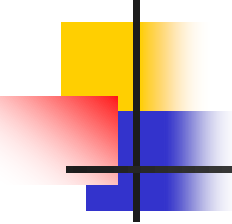
Описание на грешката в проект 3

- Трябва да се посочи кой трябва да бъде информиран за появилото се несъответствие**
- като от него се очаква да поеме отговорността за отстраняването.
 - Следва да се **даде кратко описание** на възникналия проблем по уникален начин, който да съдържа неговата същност.
 - Описват се стъпки за възпроизвеждане на несъответствието.
 - Описва се полученият резултат и се посочват разликите с планирания такъв.
 - Могат да се приложат прикачени файлове/свалени екрани, които да улесняват изясняване причините за грешката.



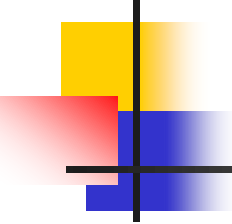
Цел на измерване на качеството

- Основните цели на софтуерните метрики на качеството са:
 - Да се даде механизъм за оценка на текущото състояние
 - Да се оценят процесите и дейностите, свързани със създаването на софтуер
 - За улесни управлението, както и планирането
 - Да идентифицира ситуации, които изискват или позволяват подобряване на процеса на разработка или процеса на поддръжка под формата на превантивни или коригиращи дейности



Имплементиране на софтуерните метрики на качеството

- Дефиниране на софтуерните метрики на качеството
- Непрекъснато прилагане
- Статистически анализ на събраните данни
- Последващи дейности
 - Промени в организацията и методите за разработване на софтуер
 - Промени в метриките и събираните данни
 - Прилагане на анализ на данните за планиране на коригиращи дейности за отделни единици



Дефиниране на нови софтуерни метрики на качеството

- Процес на 4 етапа
 - Дефиниране на атрибутите за измерване
 - Дефиниране на метрики за измерване
 - Дефиниране на индикатори
 - Дефиниране на процес за прилаганена метриците
 - Метод на докладване
 - Метод за събиране на данни за метриците



Ограничения на софтуерните метрики

- Ограничен бюджет заделен за необходимите ресурси за разработване на система от метрики за качество и нейното регулярно прилагане
- Човешки фактор
- Несигурност по отношение на валидността на данните



Основни фактори засягащи параметрите на процеса

- Стил на програмиране
- Обем на коментари в изходния код
- Сложност на софтуера
- Процент на повторно използван код
- Професионализъм на екипите за преглед на дизайна и тестване на софтуера
- Стил на докладване на резултатите от преглед и тестване



Основни фактори засягащи параметрите на продукта

- Качество на инсталирания софтуер и неговата документация
- Стил на програмиране и обем на коментари в изходния код
- Сложност на софтуера
- Процент на повторно използван код
- Брой инсталации, брой на потребителите



Основни въпроси пред софтуерните организации

- Как да определят факторите за качество и атрибутите, които ще се измерват;
- Какви трябва да бъдат стойностите на атрибутите, за да се достигне желаното качество;
- Как да се измерват определените атрибути, какви инструменти да се използват за целта;
- Какви анализи могат да се направят на тази база, свързани с определяне на текущия статус;
- Какви мерки да се изберат за подобрене на процесите и резултатите от тях.



Обобщение

- Теория на измерването
 - Основни понятия и нива на измерване
- Метрики
 - Изясняване на целите и изискванията към софтуерните метрики на качество
 - Категоризиране на метриките
 - Имплементиране и ограничения