

Факултет инжењерских наука Универзитета у Крагујевцу

Јована С. Јанићијевић
Оцена пројекта развоја
информационих система
Дипломски рад

Крагујевац, 2020.

Факултет инжењерских наука Универзитета у Крагујевцу



Назив студијског програма: Рачунарска техника и софтверско инжењерство

Ниво студија: Основне академске студије

Предмет: Пројектовање информационих система и база података

Број индекса: 560/2016

Јована С. Јанићијевић
**Оцена пројеката развоја
информационих система**
Дипломски рад

Комисија за преглед и одбрану:

1. Проф. Др Милан Ерић - ментор
2. _____
3. _____

Датум
одбране: _____

Оцена: _____

ПРИЛОГ 3: Образац пријаве завршног рада



Универзитет у Крагујевцу
Факултет инжењерских наука



Основне студије: **Рачунарска техника и софтверско инжењерство**

Име и презиме: **Јована Јанићијевић**

Број индекса: **560/2016**

ПРИЈАВА ДИПЛОМСКОГ РАДА

Тема рада: ***Оцена пројекта развоја информационих система***

Задатак: У оквиру дипломског рада кандидат треба да на конкретном примеру прикаже примену стандарда за оцену пројекта развоја информационих система са акцентом на оцени квалитета софтвера. Радом обухватити: Уводна разматрања везана за одговарајуће стандарде, захтеве, документацију, управљање итд. На крају рада дати закључна разматрања.

Крагујевац, 13. март 2020.

Ментор:

M. Erić

Проф. др Милан Д. Ерић



Изјава о самосталној изради рада

Изјављујем да сам овај дипломски рад урадила самостално, служећи се стеченим знањем и искуством као и наведеном литературом.

Јована Јанићијевић 560/2016

(потпис)

Садржај

1. УВОД.....	7
2. СОФТВЕРСКИ ПРОЈЕКТИ	8
3. ДЕФИНИЦИЈЕ КВАЛИТЕТА СОФТВЕРА.....	10
4. КВАЛИТЕТ СОФТВЕРА	11
4.1 ФАЗЕ КВАЛИТЕТА СОФТВЕРА	13
4.2 ОПШТА КЛАСИФИКАЦИЈА АТРИБУТА КВАЛИТЕТА	13
4.3 ВЕЗЕ ИЗМЕЂУ АТРИБУТА КВАЛИТЕТА	15
4.4 ПОГЛЕДИ НА КВАЛИТЕТ СОФТВЕРА	15
4.4.1 Квалитет производа.....	15
4.4.2 Квалитет процеса развоја	16
4.4.3 Квалитет са становишта окружења	16
4.5 МОДЕЛИ КВАЛИТЕТА	17
4.5.1 McCall-ов модел	18
4.5.2 Boehm-ов модел.....	22
4.5.3 Dromey-ов модел	25
4.5.4 Модел ISO 9126	26
4.5.5 CMM (Capability Maturity Model)	31
5. ФОРМИРАЊЕ ДОКУМЕНТАЦИЈЕ СИСТЕМА КВАЛИТЕТА ЗА ОБЕЗБЕЂЕЊЕ КВАЛИТЕТА СОФТВЕРА	32
6. CASE АЛАТ ЗА УПРАВЉАЊЕ ДОКУМЕНТАЦИЈОМ ЗА ОБЕЗБЕЂЕЊЕ КВАЛИТЕТА СОФТВЕРА	33
7. КОНТРОЛА КВАЛИТЕТА КРОЗ ТЕСТИРАЊЕ.....	34
8. ЗАКЉУЧАК.....	36
9. ЛИТЕРАТУРА.....	37

Резиме

Овај рад представља опис квалитета софтвера који се остварује кроз низ неопходних корака. Дефинисањем фаза квалитета, модела, формирање документације, ствара се нека врста водича која треба да обезбеди имплементацију поузданог, безбедног, ефикасног софтверског производа. Циљ овог рада јесте да се, кроз неколико појмова, смањи број неуспелих пројеката и појаве неповољних догађаја.

Кључне речи: обезбеђење квалитета, софтвер

Abstract

This document presents the description of software quality which can be achieved through several necessary steps. It consists of definitions of quality, models, documentation, which represent guide which main goal is to achieve implementation of reliable, safe and easy to use software product. The aim of this project is to reduce the number of unwanted events and unsuccessful projects.

Key words: quality assurance, software

1. УВОД

Обезбеђење квалитета софтвера представља важну ставку у свету информационих технологија и софтверској индустрији. Ако погледамо шире, можемо да приметимо да је квалитет подједнако важан у свим животним аспектима. То је један од кључних услова који води до успешног исхода. Једна од најважнијих карактеристика које квалитет носи са собом јесте сигурност. Без обзира на сложеност пројекта, значај квалитета се не смањује.

У данашње време када технологија преовладава у многи сферама, и у онима које се тичу људских живота, неопходно је поставити питање квалитета софтвера. Све се темељи на поузданости, ефикасности и провереном раду софтвера. Уколико је квалитет лош, то са собом повлачи низ других негативних последица.

При процени квалитета, није нагласак само на производу, већ и на процесу његовог развоја, примени, на коришћеним ресурсима. Тиме се утврђује испуњеност постављених функционалних захтева, степен продуктивности, перформансе система и друге нефункционалне особине.

Такође, софтверска индустрија издваја велике количине новца како би се задовољили захтеви тржишта. Из свега наведеног се може увидети колика је важност квалитета, и да само имплементација система квалитета и обезбеђење квалитета софтвера може донети позитивне резултате.

Управо због објашњених проблема и потреба, циљ овог рада јесте да дефинише пут за обезбеђивање квалитета софтвера кроз дефинисање модела, пратеће документације система квалитета, фаза и контрола квалитета кроз тестирање.

2. СОФТВЕРСКИ ПРОЈЕКТИ

У свету се годишње, у софтверске пројекте, уложи више милијарди долара. Према Standish Grupi [5], годишње се у САД реализује преко 175 000 софтверских пројеката. Међутим, велики број ових пројеката се заврши неуспехом. На слици 1 дат је приказ резултата софтверских пројеката, где:

- Групу 1 чине успешни пројекти, који су завршени у складу са предвиђеним роком, у оквиру планираног буџета и са свим функцијама и особинама које су иницијално планиране.
- Групу 2 чине пројекти који су комплетирани, али су прекорачили буџет и предвиђене рокове, и који нуде мањи број функција и особина него што је то спецификовано на почетку.
- Групу 3 чине пројекти чији је исход неуспешан, односно они пројекти који су обустављени у некој тачки развојног циклуса.



Слика 1 – Резултати софтверских пројеката

Занимљиво је погледати број и структуру пројеката који су прекорачили дефинисане временске рокове и дефинисана финансијска средства [5], који су дати у табелама 1 и 2. Подаци приказани на слици 1, као и подаци приказани у табелама, указују да се значајна финансијска средства губе када се прекорачи време и буџет код софтверских пројеката, а да се велики број софтверских пројеката никад и не заврши.

Једна од главних последица које се јављају, услед ових приказа, јесте велики број отказа и грешака код готових софтверских решења. У зависности од броја грешака које се јављају, у табели 3 су дати случајеви појаве отказа код готових софтверских производа, поредећи резултате из 1995. и 2000. године.

Табела 1 – Прекорачење времена
код софтверских пројеката

Prekoračenje vremena	% Projekata
Ispod 20%	13.9%
21 - 50%	18.3%
51 - 100%	20.0%
101 - 200%	35.5%
201 - 400%	11.2%
Preko 400%	1.1%

Табела 2 – Прекорачење буџета
код софтверских пројеката

Prekoračenje budžeta	% Projekata
Ispod 20%	15.5%
21 - 50%	31.5%
51 - 100%	29.6%
101 - 200%	10.2%
201 - 400%	8.8%
Preko 400%	4.4%

Табела 3 – Случајеви појаве отказа код готових софтверских производа

	Rezultati iz 2000	Rezultati iz 1995
Značajan broj grešaka	27%	17%
Više grešaka	21%	29%
Bez promene	11%	23%
Manje grešaka	19%	23%
Značajno manje grešaka	22%	8%

Иако су технологија и алати за развој софтвера напредовали последњих година, и даље је присутан значајан број грешака и отказа који се јављају код готових софтверских производа. Боља реализација пројеката развоја софтвера и квалитетнији софтверски производи могу се добити кроз:

- Унапређење методологија управљања пројектима и имплементација старих и унапређених решења на проблеме управљања софтверским пројектима.
- Имплементацију и развијање стандарда система квалитета који ће се примењивати на софтверске производе.

Обезбеђење квалитета представља једну од водећих парадигми модерног пословања. Посебан проблем представља дефинисање метрике софтвера, односно оног типа мерења који се односи на софтверски систем, процес или припадајућу документацију. Нарочито је битно да се развој софтвера одвија у складу са унапред дефинисаним, стандардно прописаним процедурама како би се умањиле интервенције на поправци и измени софтверских решења у каснијим фазама у току животног циклуса софтверског производа.

3. ДЕФИНИЦИЈЕ КВАЛИТЕТА СОФТВЕРА

Са становишта софтверског инжењерства, квалитет је један од неколико важних фактора, укључујући цену, планирање и функционалност. Свака дефиниција о софтверском квалитету би требало да буде оријентисана ка потребама корисника. Приликом дефиниције квалитета треба посматрати софтвер из корисничког угла гледања. Кључна питања су: ко су корисници, шта им је важно и како њихови приоритети утичу на израду софтвера.

Квалитет софтвера је планирани и систематски скуп активности који обезбеђује квалитет уграђен у софтвер. Садржи поузданост, контролу и инжењеринг квалитета софтвера. Као атрибут, квалитет софтвера је степен по којем систем, компонента или процес испуњавају:

- одређене захтеве.
- потребе или очекивања корисника.

Према Prahalad-у и Krishnan-у, високо-квалитетан софтвер се може дефинисати помоћу три основна елемента: коегзистенција, прилагодљивост и иновација.

Квалитет можемо посматрати из различитих углова. Са становишта Kitchenham-а и Pfleeger-а, постоји пет погледа на квалитет:

- Трансценденталан – квалитет је тешко дефинисати или описати у кратким цртама, али се може препознати ако је присутан.
- Кориснички – квалитет је способност проналажења сврхе или способност примећивања потреба корисника.
- Поглед заснован на производњи – квалитет значи поштовање стандарда.
- Поглед заснован на производу – квалитет је фокус на нераздвојиве карактеристике у производу, у нади да контрола тим факторима резултира унапређењем производа током коришћења.
- Поглед заснован на вредностима – квалитет је спремност купца да плати софтвер.

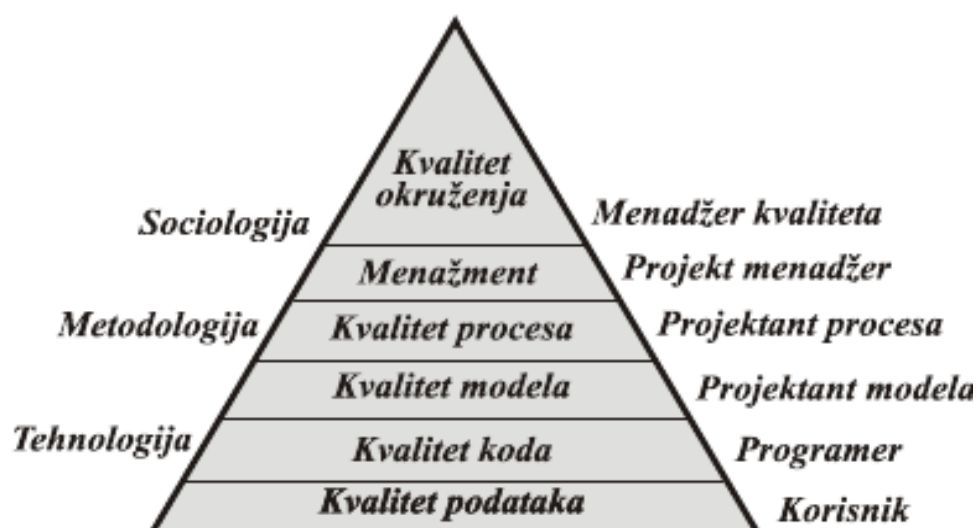
Карактеристике једног ентитета, које имају способност да задовоље формулисане и садржане потребе чине квалитет софтвера.

Квалитетан софтверски производ мора имати одређене карактеристике које су повезане са захтевима корисника и задовољавају те захтеве.

4. КВАЛИТЕТ СОФТВЕРА

Једна од идеја квалитета производа је тежња да тај производ буде израђен у складу са спецификацијом. Међутим, постоји неколико специфичности које квалитет софтвера одвајају од квалитета осталих производа [3]:

- Спецификација сваког производа иде ка испуњавању жеља и потреба купца. Али, код развоја софтвера често постоје захтеви који нису изричито специфични.
- Не постоји поуздан начин за квантификацију појединих карактеристика софтвера, као што су, на пример, стабилност, поузданост.
- Софтверска спецификација је често некомплетна.



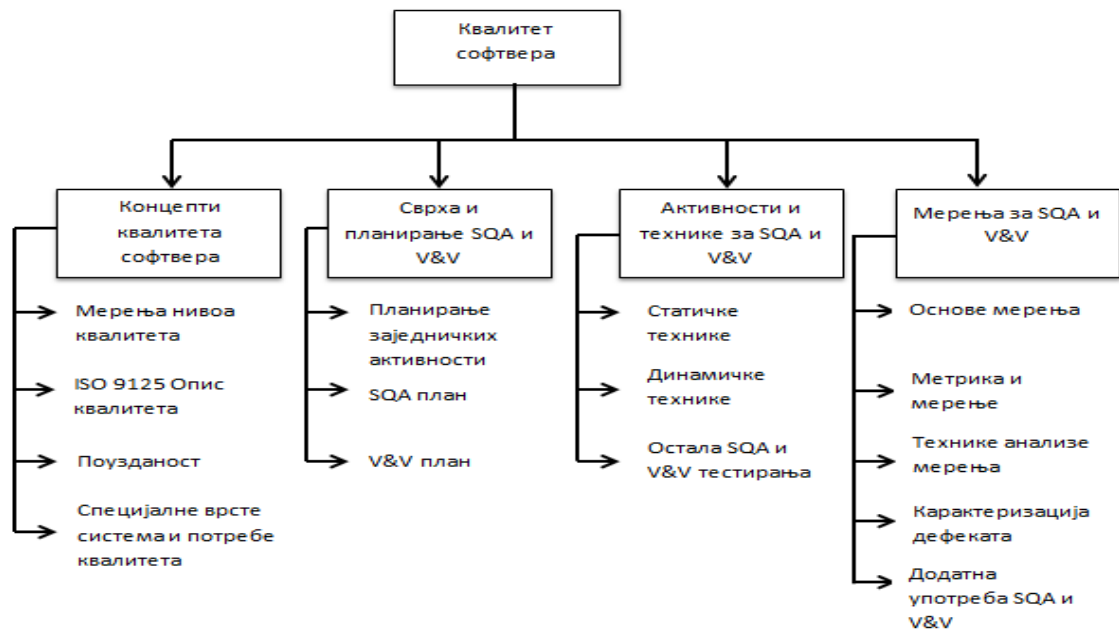
Слика 2 – Контекст квалитета софтвера

Стога је код менаџмента квалитетом софтвера неопходно водити рачуна о следећим активностима:

- Обезбеђивању квалитета, који осигурава да организација прати циљеве квалитета, што укључује дефинисање и избор различитих стандарда који се укључују у процес развоја софтвера или сам софтверски производ. Изабрани стандарди могу бити ISO 9000 или JUS/ISO 12207 – Информациона технологија – процеси животног циклуса софтвера; и/или неки национални стандарди, као на пример BS 5750. Свако предузеће према изабраном систему квалитета креира свој пословник о квалитету из кога проистиче документација система квалитета која обезбеђује квалитет одвијања одговарајућих процеса у организацији и

служи за квалитетно вођење различитих пројеката, међу којима су и пројекти развоја софтвера.

- Контрола квалитета обезбеђује да се процес развоја софтвера одвија према дефинисаним и изабраним стандардима.
- Планирању квалитета који треба да почне у раним фазама развоја софтвера. Планови квалитета треба да дефинишу жељени сет карактеристика које софтверски производ треба да има.



Слика 3 – Концепт квалитета софтвера

Уколико се посматра у ужем смислу, софтвер је алат који се користи унутар неког система, али са друге стране, софтвер значајно утиче на функционисање тог система. С обзиром на то да софтвер може у великој мери утицати на рад и функционисање пословног или неког другог система, потребно је да софтвер задовољава многобројне атрибуте квалитета и да буде квалитетан у целини.

Софтвер, на првом месту, мора да задовољи захтеве корисника и да поседује одређен ниво квалитета, а не само функционалност. Квалитет података се може специфирати кроз:

- Тачност – мера одступања неке вредности v од неке друге вредности v' која се сматра тачном.
- Комплетност – степен до кога је специфична вредност укључена у колекцију података.

- Конзистенција – дефинише се на нивоу презентације, вредности и изгледа.
- Интерпретација – формат у коме је податак специфициран и јасно дефинисан.
- Поузданост – дефинише се кроз поузданост података и поузданост извора.
- Правовременост – мера колико је податак правовремен и одговарајући за неку операцију.

Као што је потребно мерити и вредновати ове карактеристике да би се дефинисао квалитет података, потребно је мерити и многе друге карактеристике које нам омогућавају утврђивање квалитета софтвера, при чему метрика софтвера има значајну улогу. Из тог разлога се морају дефинисати карактеристике, које, углавном, нису тако експлицитне, али је њихов утицај на перформансе и квалитет софтвера врло важан.

4.1 ФАЗЕ КВАЛИТЕТА СОФТВЕРА

Из перспективе развоја софтвера, квалитет чине пет фаза [1]:

- Фаза 1 – представља основни код квалитета: синтаксно и кодно конструисање.
- Фаза 2 – представља детаљан план, опис посла: логичко конструисање програма и акција потребних да би ови програми изводили функције, односно циљеве, због којих су и конструисани.
- Фаза 3 – представља план на вишем нивоу: системска питања око интерфејса, компатибилности, учинка, безбедности и сигурности.
- Фаза 4 – представља фокусирање над захтевима: одређивање и дефинисање захтева са посебним освртом на закључивање шта се тражи између редова.
- Фаза 5 – представља осврт на корисника и шта је потребно урадити да би им се обезбедио заиста квалитетан производ. Иако је наведена као последња фаза, потребе корисника морају имати највећи приоритет.

4.2 ОПШТА КЛАСИФИКАЦИЈА АТРИБУТА КВАЛИТЕТА

Класификација атрибута квалитета гласи:

- Захтеви – параметри по којима се атрибути система доносе, спецификују и оцењују.

- Атрибути као посебни фактори – особине система, као што су полисе и механизми уграђени у систем, и њихово окружење које има утицај на захтеве. У зависности од атрибута, атрибути као посебни фактори су унутрашње или спољашње одлуке које утичу на захтеве. Фактори не морају бити независни и могу да имају узрочно-последичне везе. Фактори и њихове међусобне везе требало би да буду уврштени у архитектуру система. Међу њима су:
 - Фактори учинка – аспекти система који доприносе учинку. Ово укључује захтеве из околине и одговор система на исте.
 - Оштећења поузданости – аспекти који доприносе недостатку поузданости. Постоји узрочни ланац између недостатака унутар система и пропуста посматраних у околини. Недостаци узрокују настанак грешака. Грешка је стање система које може довести до пропуста ако се не исправи.
 - Фактори заштите – аспекти система који доприносе заштити. У те карактеристике спадају систем/околина, интерфејс, карактеристике и унутрашње карактеристике, као што је темељност.
 - Оштећења сигурности – аспекти система који доприносе недостатку сигурности. Ризици су услови или стања система који могу да воде до незгода или несрећних случајева. Незгоде и несрећни случајеви су непланирани догађаји са непожељним последицама.
- Методе – како се рукује са захтевима: процеси анализе и синтезе током развоја система, процедуре и обучавања за кориснике и операторе. Методе могу бити за анализу и/или синтезу, процедуре и/или обучавања, или процедуре које се користе у току развоја или извршења.



Слика 4 – Опишта класификација атрибута квалитета

4.3 ВЕЗЕ ИЗМЕЂУ АТРИБУТА КВАЛИТЕТА

Систем често не успева да испуни захтеве, недостатак квалитета. То се дешава када се дизајнер фокусира искључиво на испуњавање одређених захтева, не узимајући у обзир друге захтеве или узимајући, али прекасно у процесу развоја.

На пример, могуће је да се истовремено не може испунити више захтева. То не представља нови проблем, и људи који се баве развојем софтвера покушавају да се изборе с тим већ дуже време. О томе је писао и Boehm: “Најзад смо дошли до закључка да прорачунавање и разумевање вредности једне целокупне метрике за квалитет софтвера може стварати више проблема него што то заслужује. Највећи проблем је тај што су многе појединачне карактеристике квалитета у конфликту; додатна ефикасност често има исту цену као преносивост, тачност, разумљивост, одрживост; додатна тачност је често у сукобу са преносивошћу преко зависности исказане величине; концизност може бити у конфликту са јасноћом. Корисници углавном сматрају да је тешко одредити приоритете у тако конфликтним ситуацијама.” [1]

4.4 ПОГЛЕДИ НА КВАЛИТЕТ СОФТВЕРА

Процена квалитета софтверског производа зависи од контекста у коме се производ посматра. Исти недостатак се не може третирали на исти начин у сваком софтверу. Недостатак у софтверу за обраду текста нема исти значај, а ни последице, као када се појави у софтверу намењеном заштити података, где може да угрози цео систем.

У зависности од гледишта, квалитет софтвера се може посматрати на три начина: кроз квалитет производа, квалитет поступка израде производа и квалитет са становишта пословног окружења у коме се софтвер користи. [2]

4.4.1 Квалитет производа

Процена квалитета производа подразумева утврђивање да ли је софтвер квалитетан или није на основу његових карактеристика. Међутим, ако би се исти софтвер дао већем броју особа да процене његов квалитет, оне би при томе вероватно разматрале различите карактеристике. То је зато што свако посматра софтвер са свог становишта.

За корисника, софтвер је квалитетан ако има потребну функционалност, једноставну обуку и лако се користи. Може се рећи да корисник посматра софтвер споља, тј. са аспекта употребе. Када мери квалитет софтвера, корисник процењује број отказа, као и врсте отказа. Често отказе класификује у минорне, главне и отказе са катастрофалним последицама. Што је већи број катастрофалних и главних отказа, софтвер је неквалитетнији.

Софтвер процењују и пројектанти и други учесници у развоју, посебно тим за одржавање који треба да настави са радом на софтверу. Они обично анализирају интерне карактеристике софтверског производа, тј. посматрају софтвер изнутра. То подразумева процену недостатака у захтевима, у пројекту, у програмском коду. Учесници у развоју мере квалитет софтвера, такође, кроз његове недостатке, али су ти недостаци другачије природе од недостатака које мери корисник.

Сваки од наведена два погледа на софтвер има свој значај и не може се занемарити. Иако и корисници и пројектанти имају своје аргументе, њихови закључци се, по питању квалитета софтвера, могу разликовати. На пример, ако се неки софтвер тешко учи и користи, корисник може да га процени као лош, иако он поседује функционалност која је нова и вредна, што му свакако представља квалитет и даје му предност у односу на конкурентске софтвере.

Да би се постигла јединствена и објективна оцена квалитета софтверског производа, потребно је објединити спољашњи поглед корисника са унутрашњим погледом учесника у развоју софтвера. Овај циљ је постигнут увођењем модела квалитета, о чему ће бити више речи у следећем поглављу 4.5.

4.4.2 Квалитет процеса развоја

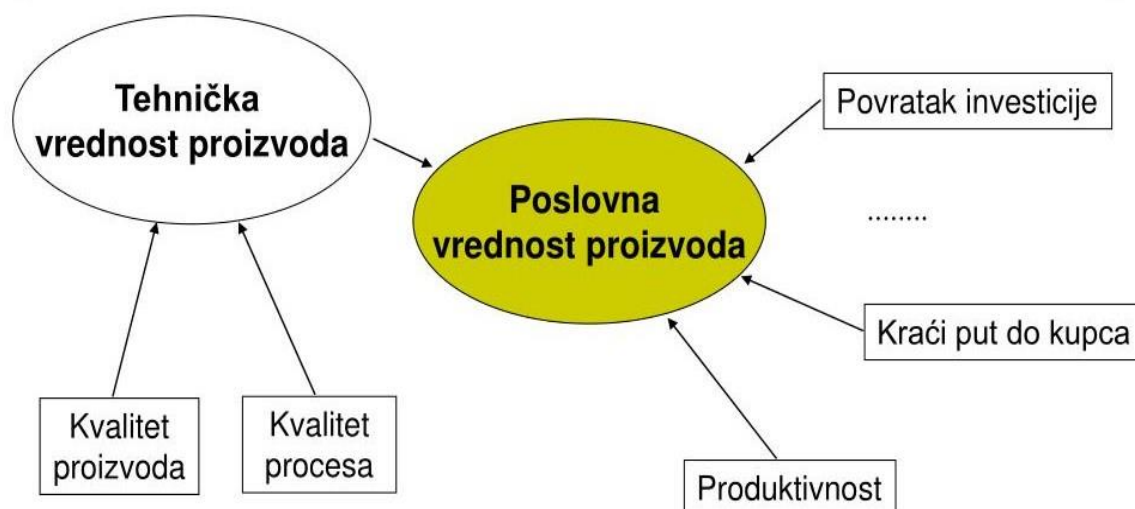
Квалитет процеса развоја софтвера је подједнако важан као и квалитет производа. Током развоја, извршавају се многе активности које имају утицаја на квалитет финалног производа. Ако нека од активности крене у погрешном смеру, смањује се квалитет произведеног софтвера.

Процес моделовања развоја софтвера знатно доприноси бољем квалитету софтвера. Моделовањем је омогућена правовремена анализа система, уз уочавање и отклањање постојећих недостатака и налажење начина за даља унапређења система. Током моделовања, значајно време се троши на анализу недостатака, што директно утиче на квалитет. Ова анализа даје одговоре на питања где и када ће се вероватно појавити недостаци, како их што пре пронаћи, да ли се може уградити толеранција на грешке како оне не би прешле у отказ, и слично. Анализи могу бити подвргнуте све фазе у процесу развоја.

4.4.3 Квалитет са становишта окружења

Сваки софтвер намењен је коришћењу у датом пословном окружењу. Уколико то није случај, сматра се да софтвер нема смисла. Након испоруке, софтвер улази у фазу употребе, уз повремено одржавање. То уводи још један поглед на квалитет софтвера, а то је поглед са аспекта пословања. За разлику од оцењивања квалитета производа и процеса развоја, који се мере бројем отказа, временом потребним за њихово отклањање, квалитет са аспекта пословања се посматра у зависности од производа и услуга које пружа окружење у ком софтвер ради.

Имајући ово у виду, може се сматрати да софтверски производ поседује своју техничку и пословну вредност. Техничку вредност чине карактеристике производа и процеса развоја. Оне зависе од начина израде софтвера, односно од техничког аспекта. С друге стране, пословна вредност представља пословне ефекте које софтвер производи у радном окружењу. То су економске категорије, као на пример продуктивност, трошкови, период повратка инвестиције. Ове две вредности су међусобно зависне. Повећање техничке вредности производа свакако утиче на пословну вредност. Однос техничке и пословне вредности производа приказан је на слици 5.



Слика 5 – Техничка и пословна вредност производа

Техничка вредност производа је важна за директне кориснике софтвера јер утиче на њихово прихватање софтвера. За руководство компаније, битнија је пословна вредност. Они квалитет мере кроз остварену добит, која је остварена захваљујући увођењу софтвера, трошкове одржавања, трошкове измена, однос времена оперативног рада и застоја.

4.5 МОДЕЛИ КВАЛИТЕТА

Модели квалитета су настали због потребе укључивања различитих аспеката посматрања у процену квалитета производа. Без обзира на велики број модела квалитета описаних у литератури, сви модели подржавају исти приступ. Сваки модел дефинише скуп карактеристика производа које сматра важним. Ове карактеристике потичу како од спољашњег, тако и од унутрашњег погледа на софтвер. Међусобно су повезане на одговарајући начин и имају удела у поступку оцене квалитета.

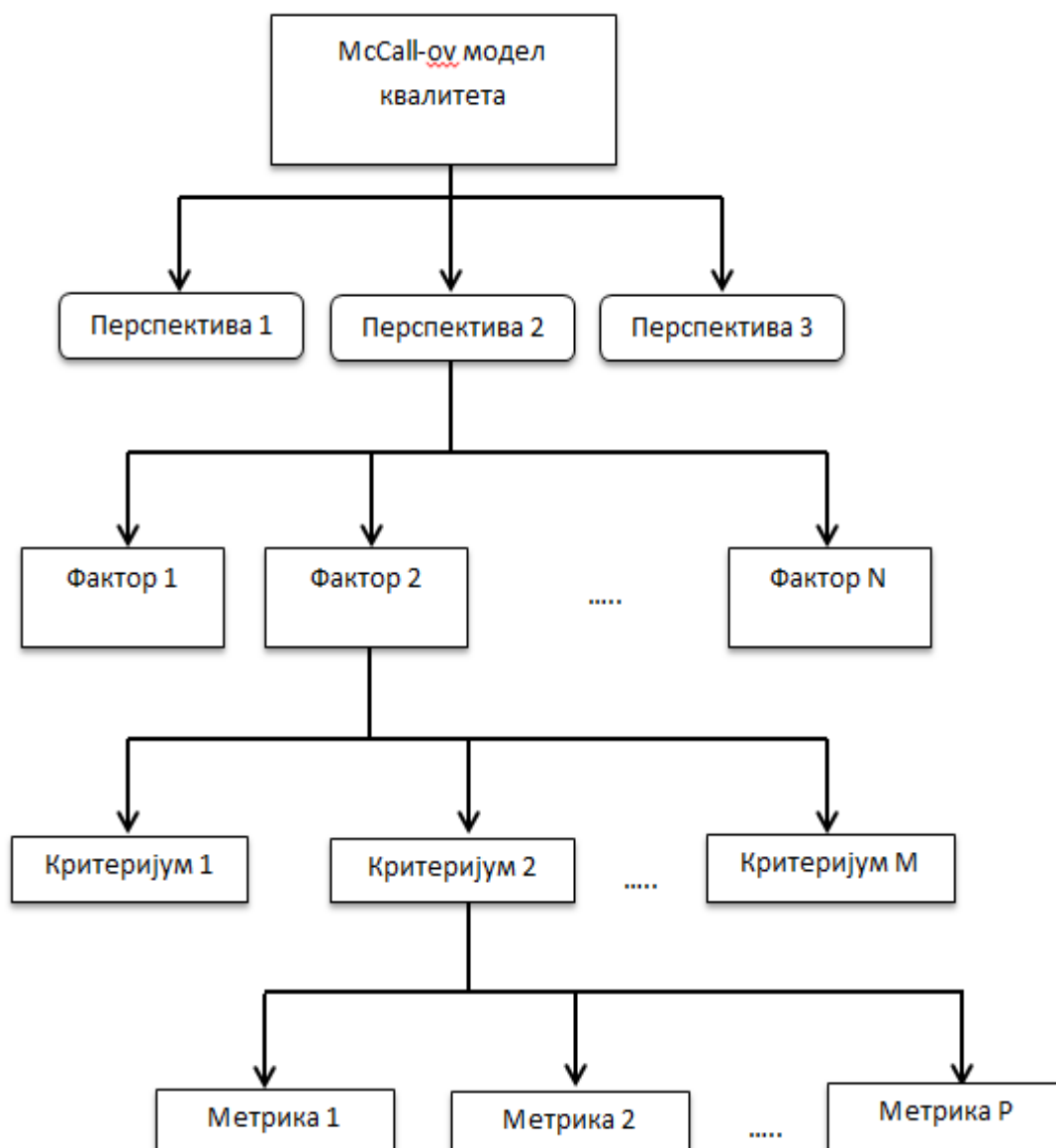
Многи модели квалитета су недоречени, тј. не прецизирају у потпуности поступак процењивања квалитета производа, већ углавном дају оквире и смернице који се могу користити у процени. Такође, не наводе се разлози зашто су поједине карактеристике укључене у модел, а друге не. Нема објашњења ни зашто се нека карактеристика налази баш на конкретном месту у моделу, а не на неком другом месту.

У наредним поглављима биће описани најпознатији модели квалитета: McCall-ов модел, Boehm-ов модел, Dromey-ов модел, ISO 9126 и CMM модел.

4.5.1 McCall-ov модел

McCall-ov модел је један од најпознатијих и најстаријих модела. Предложио га је McCall 1977. године и изворно је био намењен америчкој војсци. Модел представља покушај превазилажења разлика у схватањима корисника и развојног тима по питању квалитета софтвера.

McCall-ov модел чини хијерархија фактора, критеријума и метрике којима се производ описује, како са спољашњег, тако и са унутрашњег аспекта. Фактори представљају различите врсте карактеристика које описују понашање система. Сваки критеријум се може укључити у једну или више метрика. Метрика омогућава мерење квалитета, као и управљање истим. Модел обухвата три перспективе, а приказ је дат на слици 6.



Слика 6 – Структура McCall-овог модела квалитета

У моделу постоји једанаест фактора који дефинишу квалитет производа са становишта корисника и двадесет и три критеријума који описују квалитет са становишта оних који развијају систем. Сваки фактор је повезан са два до пет критеријума. У зависности од значења, фактори су распоређени у три перспективе:

- Оперативност производа (Перспектива 1 на слици 6) – садржи факторе који се односе на употребу производа.
- Ревизија производа (Перспектива 2 на слици 6) – садржи факторе који описују могућности мењања производа.
- Транзиција производа (Перспектива 3) – садржи факторе о прилагодљивости производа раду у новом окружењу.

Перспектива оперативност производа садржи следеће факторе:

- Исправност – показује у којој мери функционалност система одговара спецификацији.
- Поузданост – указује на учесталост појаве грешака.
- Ефикасност – односи се на начин коришћења ресурса и укључује ефикасност извршавања, као и ефикасност памћења података.
- Интегритет – указује на способност заштите од неовлашћеног приступа софтверу.
- Употребљивост – описује лакоћу коришћења софтвера.

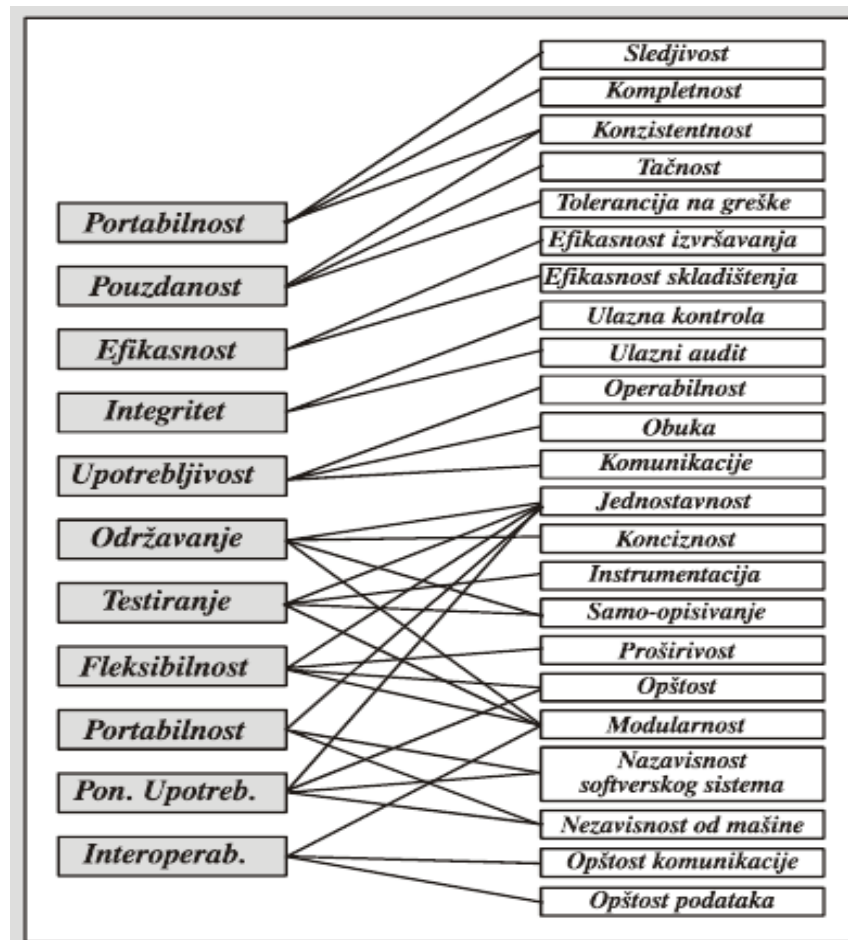
Перспектива ревизија производа обухвата:

- Могућност одржавања – указује на труд потребан за лоцирање и отклањање грешака у програму унутар радног окружења.
- Флексибилност – показује лакоћу извођења измена потребних због промена у радном окружењу.
- Могућност тестирања – односи се на лакоћу тестирања програма.

Перспективу транзиција производа чине следећи фактори:

- Преносивост – указује на напор потребан за пренос софтвера из једног у друго окружење.
- Могућност поновног коришћења – описује могућност поновног коришћења софтвера у другом контексту.

- Интероперабилност – односи се на могућности спрезања софтвера са другим системима.



Слика 7 – McCall-ov модел квалитета софтвера

У сликама испод наведени су критеријуми за факторе перспективе оперативност производа и перспективе ревизија производа.

Perspektiva	Faktor	Kriterijum
operativnost proizvoda	ispravnost	pracenje i razumljivost
		potpunost
		konzistentnost
	pouzdanost	konzistentnost
		preciznost
		tolerancija na greske
	efikasnost	efikasnost izvrsavanja
		efikasnost pamcenja
	integritet	kontrola pristupa
		revizija pristupa
	upotrebljivost	operativnost
		obuka
		komunikacija

Слика 8 – Перпсектива оперативност производа

Perspektiva	Faktor	Kriterijum
revizija proizvoda	mogucnost odrzavanja	jednostavnost
		konciznost
		samoopisivost
		modularnost
	fleksibilnost	samoopisivost
		prosirivost
		opstost
	mogucnost testiranja	jednostavnost
		instrumentacija
		samoopisivost
		modularnost

Слика 9 – Перспектива ревизија производа

Основни циљ овог модела је формирање структуре фактора и критеријума која даје потпуну слику о квалитету софтвера и примена метрика на ту структуру ради процене квалитета софтвера.

McCall дефинише скуп метрика у виду израза за рачунање утицаја појединачних елемената на квалитет. Општи облик израза за факторе квалитета је:

$$F_q = c_1m_1 + c_2m_2 + \dots + c_nm_n$$

где је F_q фактор квалитета, m_i критеријуми од којих фактор зависи, а c_i коефицијенти утицаја (тежински коефицијенти). Вредност коефицијената утицаја зависи од конкретног софтверског производа. Природа многих критеријума је таква да се њихова испуњеност може проценити само субјективно. Скуп метрика се може дефинисати и у виду табеле са означеним зависностима између фактора критеријума и критеријума битних за тај фактор. Део скупа приказан је у табели 4.

Табела 4 – Веза фактора и критеријума у метрици

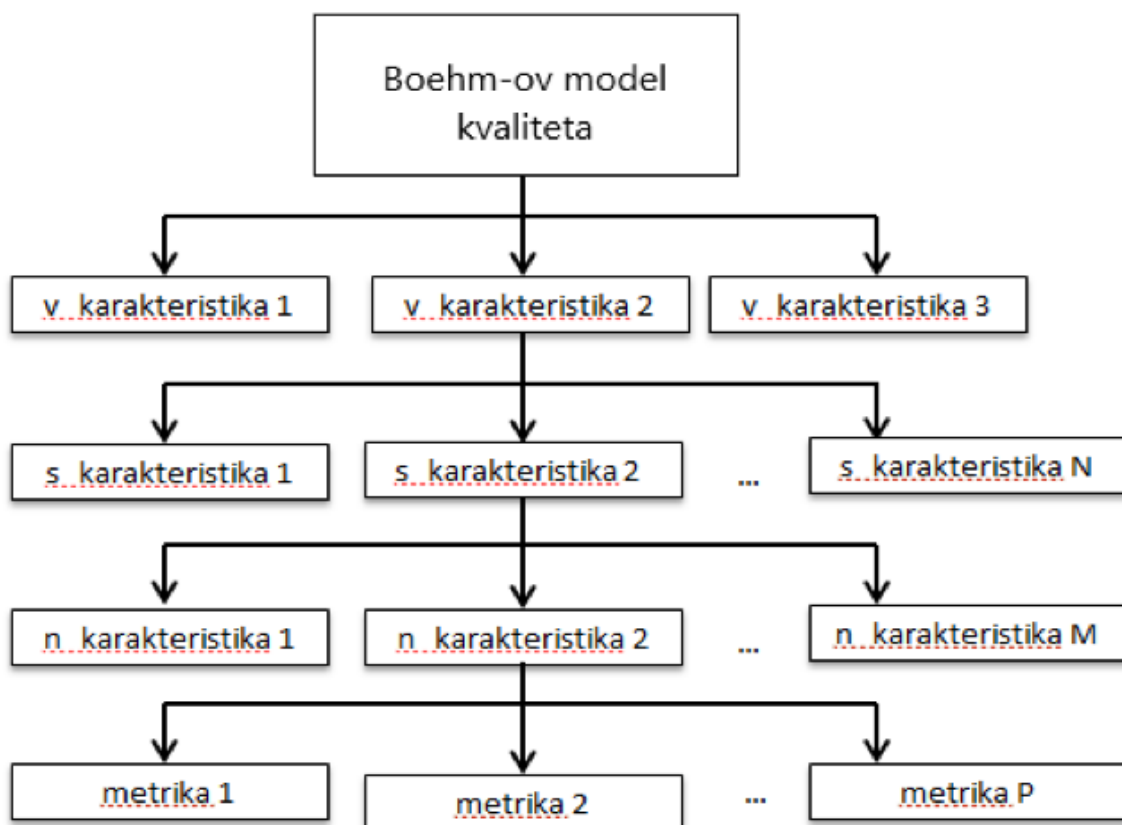
Критеријуми	Фактори	исправност	Поузданост	Ефикасност	Интегритет	Употребљивост	Одржавање	Флексибилност	Тестирање	Преносивост	Пон.коришћење	Интеропераблино
једноставност			X				X	X	X			
конзистентност		X	X				X	X				
општост								X		X	X	X
модуларност			X				X	X	X	X	X	X
толеранција на грешке			X									

При мерењу kvaliteta softvera koristi se metod odgovaranja sa „да“ и „не“ на постављена питања. На пример, да би се утврдио утицај неког критеријума, поставља се већи број питања у вези с њим. Ако је број одговора „да“ и „не“ исти, сматра се да се критеријум достиже са 50%. Метрике се могу формирати по критеријумима, факторима или по производу.

4.5.2 Boehm-ov модел

Овај модел је настао непосредно после McCall-ovog модела, 1978. године. И овај модел описује квалитет производа у виду хијерархије карактеристика које укључују очекивања и корисника и учесника у развоју. Међутим, овај модел има шири приступ од McCall-ovog зато што укључује и трошкове и ефикасност одржавања.

За све ауторе модела, примарна карактеристика квалитета је општа корисност. Софтвер, пре свега, мора бити користан. Ако то није случај, онда је његов развој окарактерисан као губљење времена, новца и рада. Сам модел се састоји од три нивоа: високог, средњег и ниског нивоа карактеристика. На дну хијерархије налазе се метрике придружене карактеристикама најнижег нивоа, као што се може видети на слици 10. [2]



Слика 10 – Структура Boehm-ovog модела квалитета

Висок ниво карактеристика описује основне захтеве по питању квалитета који морају да буду задовољени. Овај ниво садржи три карактеристике:

- Употребљивост (v karakteristika 1) – показује колико ефикасно, лако и поуздано се софтвер може користити, онакав какав је тренутно.
- Могућност одржавања (v karakteristika 2) – указује на могућност лаког разумевања, модификовања и поновног тестирања софтвера.
- Преносивост (v karakteristika 3) – односи се на могућност коришћења софтвера у случају измењеног окружења.

На средњем нивоу се налази седам карактеристика које описују шта се по питању квалитета очекује од софтверског система. То су:

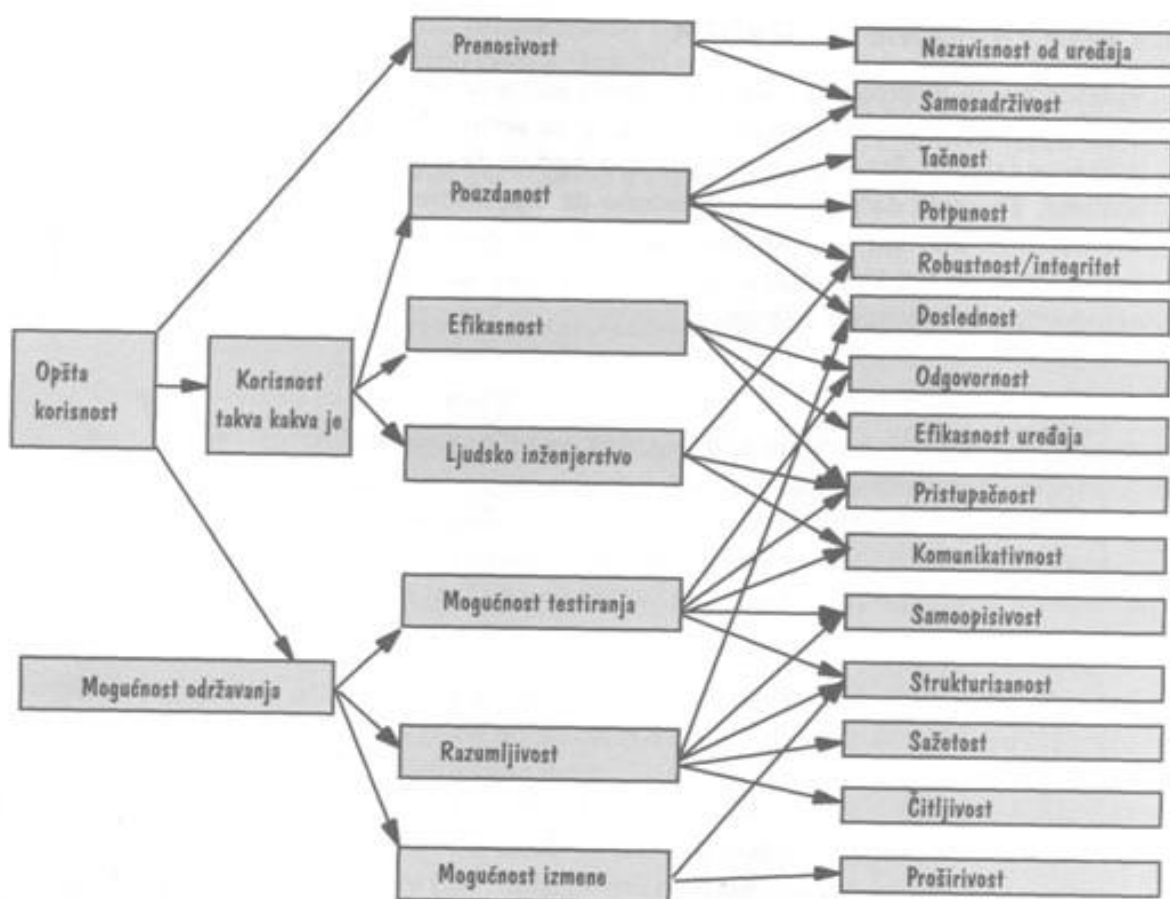
- Преносивост – софтвер може добро да ради у рачунарском окружењу другачијем од постојећег.
- Поузданост – софтвер на задовољавајући начин извршава функционалност која се од њега очекује.
- Ефикасност – систем ефикасно користи ресурсе.
- Употребљивост – софтвер је поуздан, ефикасан и погодан за коришћење.
- Могућност тестирања – успостављени су критеријуми валидације и евалуација перформанси.
- Разумљивост – сврха софтвера је јасно назначена.
- Флексибилност – софтвер омогућава лако укључивање измена у систем.

Ниски ниво обухвата петнаест карактеристика које су повезане са карактеристикама средњег и високог нивоа, на начин који је приказан на слици 11.

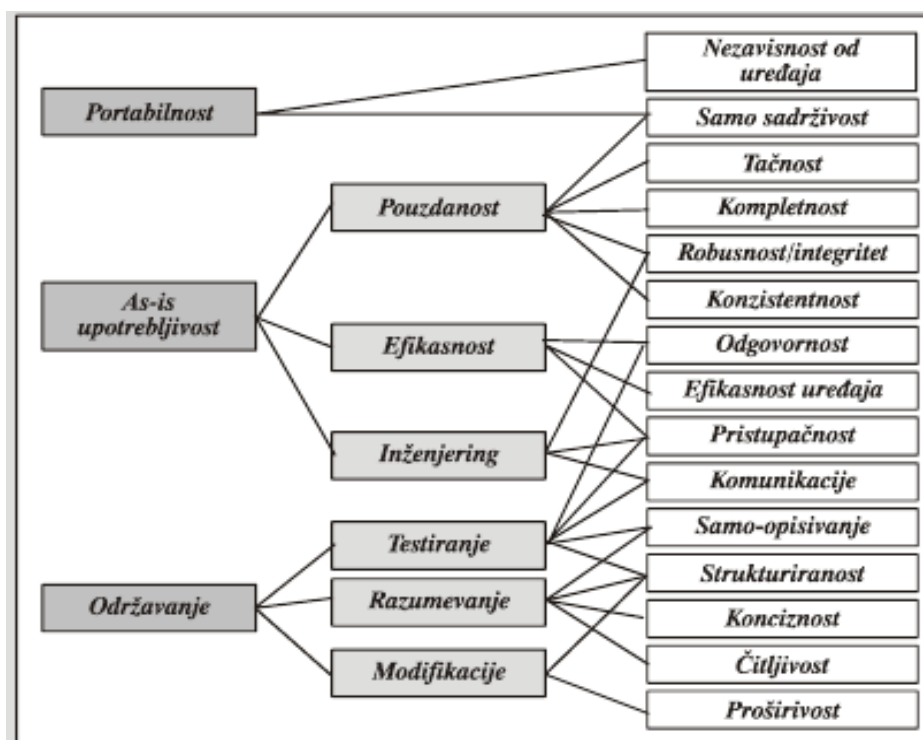
Као што се може видети, за разлику од McCall-ovog модела, Boehm-ov модел представља хијерархију која на највишем нивоу има карактеристике битне за крајњег корисника, а на дну карактеристике техничког типа. Карактеристике на најнижем нивоу представљају основу за дефинисање метрика квалитета.

Под метриком квалитета, Boehm подразумева меру обима или степена до кога софтверски производ поседује неку карактеристику квалитета. Мерљиве карактеристике се односе на високо техничке детаље квалитета које тешко могу да разумеју не-технички учесници у пројекту.

Овај модел настоји да укључи опште карактеристике квалитета у критеријуме квалитета кроз три угла посматрања, према врсти и положају корисника: крајњи корисници, потенцијални корисници на другим местима (портабилност) и потенцијалних корисника у неком другом времену. Приказ је дат на слици 12.



Слика 11 – Садржај Воехт-овог модела квалитета

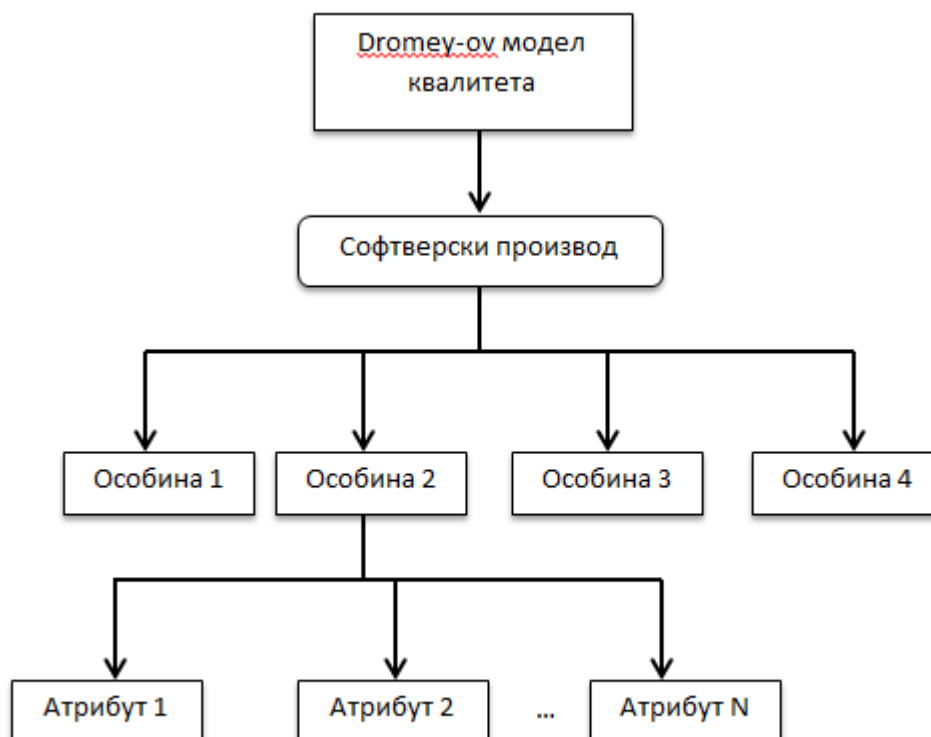


Слика 12 – Воехт-ов модел квалитета софтвера

4.5.3 Dromey-ov модел

Dromey-ov модел квалитета потиче из 1995. године. Он се заснива на производу. Уочено је да поступак процене квалитета треба да се разликује од производа до производа. Стога је потребно развити више идеја како би се оне могле применити на различите системе.

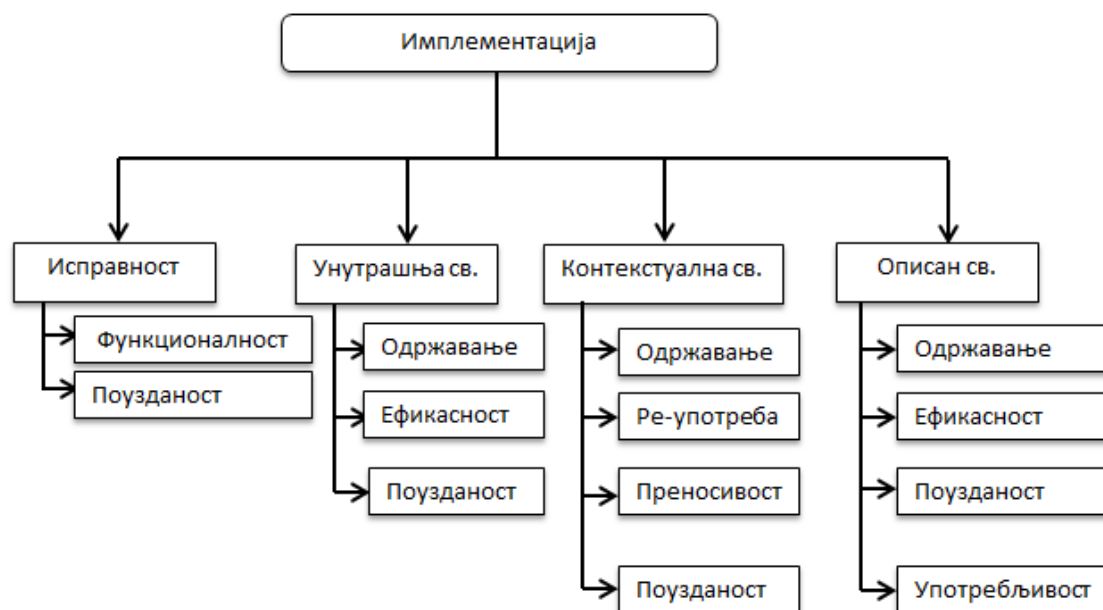
Модел квалитета производа се састоји од четири његове особине. За сваку особину, дефинисан је одређен број атрибута квалитета. Општа структура овог модела дата је на слици 13. [2]



Слика 13 – Структура Dromey-овог модела квалитета

Dromey је предложио генеричку технику за изградњу модела квалитета. По њему, квалитет производа зависи од избора компонената од којих је производ начињен, особина тих компонената и њихове композиције. По питању компонената, Dromey је развио различите моделе који се могу користити за евалуацију квалитета захтева, дизајна и самог софтвера. Идеја је била да се током целог животног циклуса производе елементи квалитета, тако да се на крају добије финални производ који испољава атрибуте високог квалитета.

Издвојене су четири особине производа: исправност и унутрашња, контекстуална и описна својства. Особине су повезане са атрибутима квалитета високог нивоа, тј. атрибутима са највећим приоритетом. Приоритети атрибута се разликују у различитим пројектима. Приказ је дат на слици 14.



Слика 14 – Садржај Dromey-овог модела квалитета

Примена модела обухвата следећих пет корака:

- Изабрати скуп атрибута квалитета високог нивоа.
- Идентификовати компоненте система.
- Идентификовати најзначајнија својства која доприносе квалитету сваке компоненте.
- Одредити како свако својство утиче на атрибуте квалитета.
- Урадити евалуацију модела и идентификовати његове недостатке.

4.5.4 Модел ISO 9126

Модел ISO (International Standardization Organization) 9126 из 1991. године настао је као резултат покушаја да се у софтверском инжењерству успостави консензус по питању терминологије везане за процену квалитета софтвера. Касније су се појавиле бројне проширене верзије овог стандарда.

ISO 9126 модел, осим модела квалитета, дефинише и низ упутстава за мерење карактеристика квалитета које се појављују у моделу.

ISO модел указује на три аспекта квалитета софтвера [4]:

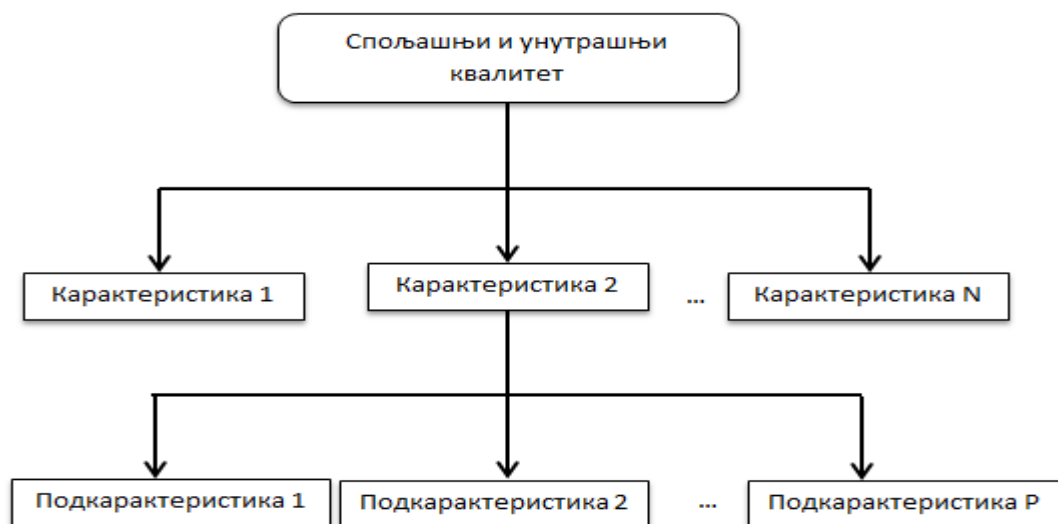
- Квалитет са аспекта употребе – представља кориснички поглед на квалитет софтвера када је софтвер у употреби у датом окружењу и датом контексту. Њиме се мери обим у коме корисници могу да остваре своје циљеве у датом окружењу, не водећи рачуна о особинама самог софтвера.
- Спољашњи квалитет – одређен је укупним спољашњим карактеристикама софтвера који се извршава. Обично се мери током тестирања у симулираном окружењу са симулираним подацима применом спољашњих метрика.
- Унутрашњи квалитет – одређен је укупним унутрашњим карактеристикама софтвера. Квалитет софтвера се може побољшати током кодирања, прегледања и тестирања.

Наведени аспекти су међусобно повезани. Атрибути битни за унутрашњи квалитет утичу на атрибуте спољашњег квалитета, а они, даље, утичу на атрибуте квалитета са аспекта употребе.

ISO 9126 се може сматрати дводелним моделом квалитета, кога чине следећи делови:

- Спољашњи и унутрашњи модели квалитета
- Модел квалитета употребе

Спољашњи и унутрашњи модел квалитета се заснива на McCall-ovom и Boehm-ovom моделу. То је модел кога чине три нивоа. Први ниво садржи шест карактеристика квалитета, други садржи двадесет и седам подкарактеристика квалитета, а трећи мере квалитета. Модел је приказан на слици 15.



Слика 15 – ISO 9126 модел за спољашњи и унутрашњи квалитет

У оквиру стандарда, предложено је више од сто мера спољашњих и унутрашњих карактеристика, мада тај скуп није потпун, већ се могу користити и друге мере. Ове мере су описане у више докумената.

У наставку је описан садржај модела, тј. значење свих његових карактеристика и подкарактеристика:

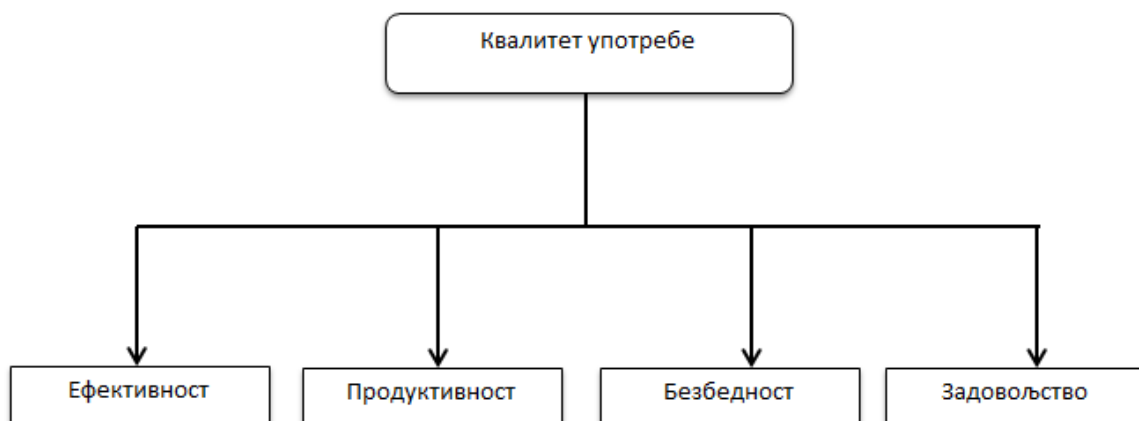
- Функционалност – способност софтверског производа да обезбеди потребне функције при раду у задатим условима. Садржи следеће подкарактеристике:
 - Погодност – способност софтвера да обезбеди одговарајући скуп функција за специфициране задатке и корисничке циљеве.
 - Прецизност – способност софтвера да обезбеди праве и договорене резултате или ефекте са потребним степеном прецизности.
 - Сигурност – способност софтвера да заштити информације и податке тако да неауторизоване особе или системи не могу да им приступају, а ауторизовани могу.
 - Интероперабилност – способност софтвера да оствари интеракцију са другим системима.
 - Функционална усаглашеност – способност софтверског производа да поштује стандарде, договоре или законске регулативе повезане са функционалношћу.
- Поузданост – способност софтверског производа да постигне задати ниво перформанси када се користи у задатим условима. Садржи следеће подкарактеристике:
 - Зрелост – способност софтвера да превазиђе недостатак настао као резултат грешке у софтверу.
 - Толеранција на грешке - способност софтвера да одржи потребан ниво перформанси у случају појаве софтверских грешака.
 - Могућност опоравка – способност софтвера да поново успостави потребан ниво перформанси и опорави податке директно укључене у настанак неког проблема.
 - Усаглашеност поузданости – способност софтверског производа да поштује стандарде, договоре или законске регулативе повезане са поузданошћу.
- Употребљивост – способност софтверског производа да се лако усваја, разуме, учи и користи када се примењује под разним задатим условима. Садржи следеће подкарактеристике:

- Разумљивост – способност софтвера да кориснику омогући да лако схвати да ли је систем одговарајући, како се користи и под којим условима.
 - Могућност учења – способност софтвера да кориснику пружи могућност учења апликације.
 - Оперативност – способност софтвера да омогући кориснику да ради и контролише систем.
 - Атрактивност – способност софтвера да буде привлачан за корисника.
 - Употреба усаглашености – способност софтверског производа да поштује стандарде, договоре, упутства или законске регулативе повезане са употребљивошћу.
- Ефикасност – способност софтверског производа да пружи одговарајуће перформансе, у складу са коришћеним ресурсима, у задатим условима рада. Садржи следеће подкарактеристике:
 - Понашање у времену – способност софтвера да при извршавању функција произведе жељени одзив у задатом времену и уз предвиђене протоке, под специфичним условима рада.
 - Понашање у вези са ресурсима – способност софтвера да користи одговарајуће количине и врсте ресурса при извршавању својих функција.
 - Усаглашеност ефикасности – способност софтверског производа да поштује стандарде и конвенције повезане са ефикасношћу.
 - Могућност одржавања – способност софтверског производа да буде модификован. Модификације укључују могуће корекције, побољшања или прилагођења у складу са променама у окружењу, променама у захтевима или функционалним спецификацијама. Садржи следеће подкарактеристике:
 - Могућност анализе – способност софтвера да дијагностификује недостатке и њихове узроке, као и да идентификује делове које треба модификовати.
 - Променљивост – способност софтвера да се у њему имплементира потребна модификација.
 - Стабилност – способност софтвера да избегне неочекиване ефекте због измена у софтверу.
 - Могућност тестирања – способност софтвера да омогући валидацију након модификације софтвера.

- Усаглашеност одржавања – способност софтверског производа да поштује стандарде и конвенције повезане са одржавањем.
- Преносивост – способност софтверског производа да се пренесе из једног окружења у друго. Садржи следеће подкарактеристике:
 - Прилагодљивост – способност софтвера да се прилагоди различитим окружењима без додатних акција или средстава у односу на оне које су већ предвиђене у софтверу за ту сврху.
 - Могућност инсталирања – способност софтвера да буде инсталиран у специфицираном окружењу.
 - Коегзистенција – способност софтвера да ради у заједничком окружењу са другим независним софтверима, делећи са њима исте ресурсе.
 - Заменљивост места – способност софтвера да буде употребљен уместо другог софтвера са истом сврхом и у истом окружењу.
 - Усаглашеност преносивости – способност софтверског производа да поштује стандарде и конвенције повезане са преносивошћу.

Из описаног садржаја се може закључити да се свака подкарактеристика односи на тачно једну карактеристику, што није случај код McCall-ovog и Boehm-ovog модела. Осим тога, подкарактеристикама је описан корисников поглед на софтвер, а не интерни поглед пројектанта.

Модел квалитета употребе садржи четири карактеристике квалитета везане за употребу софтвера, као што је приказано на слици 16.

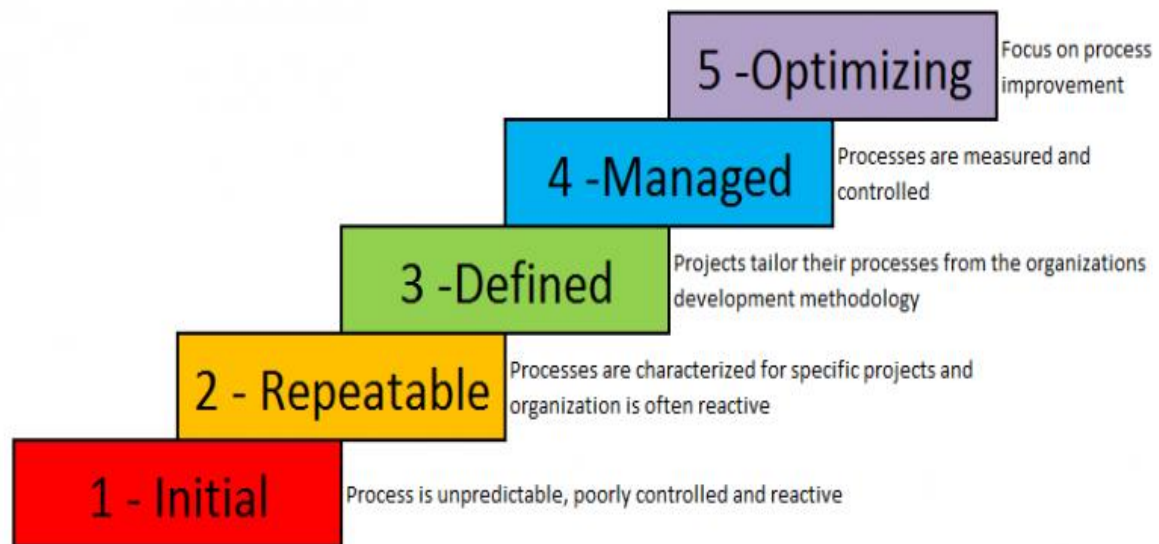


Слика 16 – ISO 9126 модел квалитета употребе

4.5.5 CMM (Capability Maturity Model)

Овај модел је настао на институту за софтверско инжењерство на Carnegie Mellon универзитету, на иницијативу Министарства одбране Сједињених Америчких Држава. СММ се састоји од пет нивоа који описују предности процеса у софтверском инжењерству. Четврти ниво овог модела се фокусира на квалитет софтвера. [1]

- У првом нивоу нема реалних процеса у целини.
- Имплементирање процеса – укључује захтеве менаџмента, планирање и праћење софтверских пројеката и формирање менаџерског тима.
- Стандардизација процеса – захтева управљање пројектом, планирање и праћење извршења пројеката помоћу организације. Приоритет се даје организационом процесу, инфраструктури, обуци.
- Четврти ниво је фокусиран на производ и процес квалитета софтвера, са могућношћу мерења квалитета при имплементацији пројекта са повратном информацијом вођама тимова које им омогућавају да врше интервенције са циљем унапређења процеса квалитета у организацији.
- Унапређење процеса – спречавају се грешке у управљању софтверско-технолошким променама, и променама у процесу управљања. Прво се, по могућству, превентивно праве промене у властитом инжењерскософтверском окружењу, ради постизања вишег нивоа ефикасности и ефикасности.



Слика 17 – СММ модел квалитета

5. ФОРМИРАЊЕ ДОКУМЕНТАЦИЈЕ СИСТЕМА КВАЛИТЕТА ЗА ОБЕЗБЕЂЕЊЕ КВАЛИТЕТА СОФТВЕРА

Један од могућих приступа у обезбеђењу квалитета софтвера је и дефинисање потребне документације система квалитета која би се односила на софтвере. Документација би требало да обезбеди да софтверски производ, који се испоручује купцу или користи у предузећу, задовољава установљене и уговорене техничке захтеве. Такође, требало би да обезбеди пријем софтверског пројекта и да обезбеди да се све политике, стандарди, искуства из праксе, процедуре и процеси примењују у току трајања пројекта по унапред утврђеном редоследу.

Неки од релевантних IEEE стандарда су коришћени да би се обухватила сва значајна места у току животног циклуса развоја производа и да би се учинило да препоручена документација за обезбеђивање квалитета софтвера буде што потпунија:

- IEEE 730 – Стандард за планове обезбеђења квалитета софтвера, прописује минимум од:
 - Спецификација захтева софтвера
 - Опис пројекта софтвера
 - План софтверске верификације и валидације
 - Извештај о софтверској верификацији и валидацији
 - Корисничка документација
 - План управљања софтверском конфигурацијом
- IEEE 828 – Стандард за планове управљања документацијом
- IEEE 829 – Стандард за документацију за софтверско тестирање
- IEEE 830 – Препоручена пракса за захтеве софтверске спецификације
- IEEE 1008 – Стандард за јединично тестирање софтвера
- IEEE 1012 – Стандард за планове софтверске верификације и валидације
- IEEE 1016 – Водич за опис софтверског пројекта
- IEEE 1028 – Стандард за софтверске контроле и аудите
- IEEE 1042 – Водич за планове управљања софтверским конфигурацијама
- IEEE 1044 – Стандард класификације софтверских аномалија
- IEEE 1045 – Стандард за метрику софтверске продуктивности
- IEEE 1058.1 – Стандард за планове управљања софтверским пројектима

- IEEE 1059 – Водич за планове софтверске верификације и валидације
- IEEE 1061 – Стандард за метрику методологије квалитета софтвера
- IEEE 1063 – Стандард за корисничку документацију софтвера
- IEEE 1074 – Стандард за развој SDLC процеса
- IEEE 1219 – Стандард за софтверско одржавање
- IEEE 1233 – Водич за развој спецификације захтева система

6. CASE АЛАТ ЗА УПРАВЉАЊЕ ДОКУМЕНТАЦИЈОМ ЗА ОБЕЗБЕЂЕЊЕ КВАЛИТЕТА СОФТВЕРА

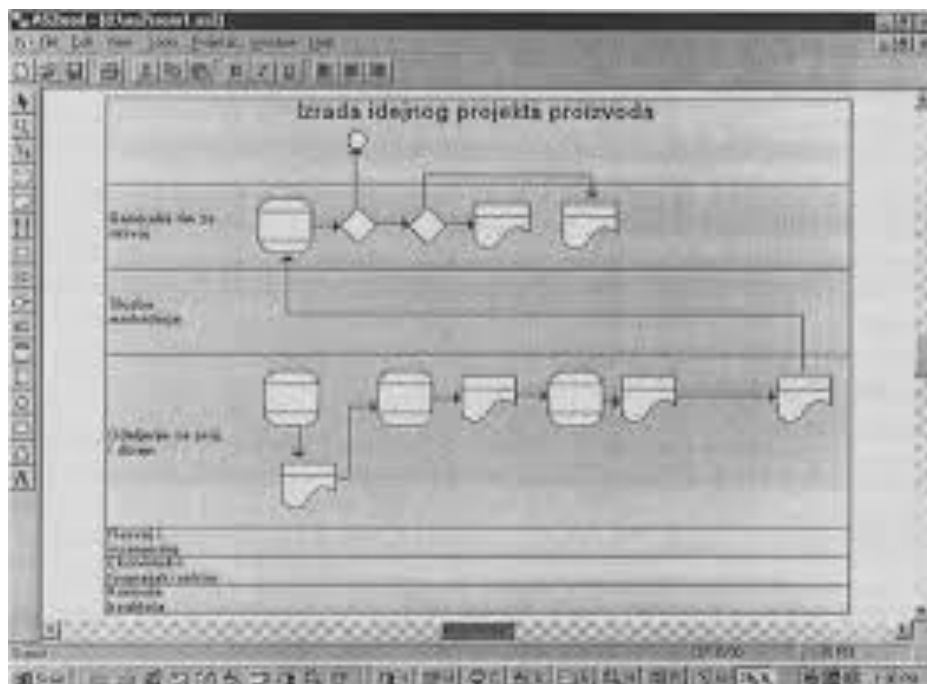
Овај алат развијен је у СИМ центру Факултета инжењерских наука у Крагујевцу [3]. Намењен је софтверским инжењерима и инжењерима система квалитета који се баве анализом и испитивањем процеса развоја софтверског производа, или неког другог производа.

Примарна замисао овог софтвера је да инжењер на једноставан и интуитиван начин за кратко време може да, путем употребљивог графичког интерфејса, прикаже процес развоја новог софтверског производа, да га анализира и у потпуности документује. У случају комплексних решења, процедура се може поделити у више међусобно повезаних процедура и инструкција. Све процедуре прате одговарајући обрасци и записи.

Основна документација, која се директно везује за активност, може бити: процедура, инструкција или упутство, што зависи од комплексности активности и организационог нивоа на коме се активности спроводе и дефинишу. Ови ентитети се могу називати поступцима. Из једног поступка може да проистекне један или више образаца. Поступци и обрасци су делови једног сценарија. Како један сценарио може да послужи као основа за развој више конкретних производа, тако и један образац може да има више својих појавних видова кроз постојање записа.

За спровођење сваке процедуре, инструкције или упутства дефинише се одговарајућа функција и одговорна особа, што се, на неки начин, може поистоветити са улогама утицајних група које су представљене на моделу обезбеђења квалитета софтвера.

Сам софтвер се састоји од неколико подцелина и свака од њих је задужена за одређену врсту трансакција. Тако постоји подцелина за израду поставке проблема, подцелина везана за документовање активности, подцелина за подешавање начина функционисања програма, подцелине за израду електронских образаца, подцелине за омогућавање приступа Интернету и Интранету. Софтвер, у оквиру свог контролног центра, омогућава поделу процеса на фазе, дефинисање функција које учествују у процесу, као и расподелу функција по појединим фазама са аспекта приказивања и анализе процеса развоја новог софтверског производа. Само окружење овог алата је стандардно Windows окружење па је врло лако за коришћење.



Слика 18 – Софтвер за управљање документацијом система квалитета

У оквиру главне форме у првој фази могуће је постављати објекте, као што су процедуре, инструкције, записе, и повезивати објекте према претходно утврђеним правилима.

У другом кораку је могуће дефинисати опште податке о објектима, на пример, ознаке, називе, затим дефинисати издања, историјат статуса и документовати све активности. Такође је омогућено прављење измена на свим поступцима и документовање тих измена.

Софтвер омогућава креирање и преглед образаца, као и потпуну манипулацију обрасцима. Развијени софтвер може послужити као врло користан алат намењен инжењерима система квалитета и може пружити значајну подршку процесу обезбеђења квалитета софтвера.

7. КОНТРОЛА КВАЛИТЕТА КРОЗ ТЕСТИРАЊЕ

Квалитет софтвера највише зависи од пројектаната и програмера, односно, вођа развојних тимова. Они дефинишу политику изградње и тестирања одређеног нивоа квалитета у различитим фазама развоја софтвера. Контрола квалитета софтвера је неопходна фаза у животном циклусу софтвера и треба је схватити као превентиву.

Појам тестирања данас се разликује од времена седамдесетих година када је то значило искључиво тражење грешака. Сада оно обухвата низ активности од планирања, дизајнирања, изградње, одржавања, па све до спровођења тестова. Потребно је

дефинисати начине мерења појединачних квалитета и дефинисати критеријуме прелазности.

Прва фаза у одређивању контроле квалитета је фаза пројектовања. За спровођење већ донете политике тестирања квалитета софтвера постоје разни методолошки прилази: структурирани, објектно-оријентисани или формално-језички оријентисани праћени одговарајућим case алатима. Тестирање је најефикаснији начин за одређивање нивоа квалитета софтверског производа. Битно је да се тестирање посматра као једна од фаза у процесу развоја софтвера. Често се на тестирање гледа као на сасвим одвојен, самосталан процес, што представља грешку. Највећи део тестирања се обавља од стране програмера који је развио софтвер, али да би тестирање било потпуно, потребно је обезбедити и тестирање од стране будућих корисника. На тај начин се могуће грешке могу најбрже пронаћи и уочити. Корисник који није упознат са начином израде софтвера, непосредном употребом ће указати програмеру на могуће грешке. Приликом тестирања није могуће испитати све акције корисника, него се треба фокусирати на мањи део акција. Из тог разлога је од велике важности одредити политику тестирања:

- Шта се тестира
- Којом брзином ће се вршити тестирање
- Колико различитих података треба уносити и сл.

Од помоћи могу бити и алати за тестирање одређених делова софтвера, али још увек није пронађен алат за тестирање целокупног софтвера.

Квалитет корисничког упутства и упутства за одржавање је мера у којој упутства одражавају стварно функционисање софтвера, као и мера у којој су информације приказане на разумљив и употребљив начин.

Циљеви тестирања су разнолики. Огледају се кроз [1]:

- Демонстрацију – проверу коректности имплементације софтвера.
- Деструкцију – детекцију грешака на нивоу програмирања.
- Евалуацију – откривање грешака у спецификацији захтева, дизајну и имплементацији.
- Превенцију – спречавању грешака кроз фазе развоја софтвера.

Иако тестирање квалитета софтвера (софтверских компоненти) даје повратну информацију о употребљивости софтвера у реалним системима, квалитет истог се не може гарантовати у потпуности због делимичне непознанице коју систем представља када се користи у реалним системима. Један од начина провере квалитета софтвера су и „бета тастери“, људи посебно обучени за тестирање прелиминарних верзија програма.

8. ЗАКЉУЧАК

На основу свега написаног, може се извести закључак да је квалитет веома битна фаза у изради производа. Није лако осигурати и обезбедити квалитет, али је потребно уложити труд, знање, нове идеје како би се исти побољшао.

Како би се остварио жељени квалитет, од великог је значаја одредити модел квалитета софтвера. У оквиру овог рада, наведено је пет модела: McCall-ov, Boehm-ov, Dromey-ov, ISO 9126 и CMM модел. Треба обратити пажњу и на све факторе и појединце за које поједине групе фактора имају битну улогу. Документација система квалитета, која се ослања на ISO и IEEE стандарде, је још једна ставка која води ка обезбеђивању бољег квалитета. Корак на који, такође, треба обратити пажњу јесте контрола квалитета кроз тестирање, где до изражаја долази политика коју одређују вође пројекта.

Када се сагледају сви поступци кроз које је потребно проћи како би се обезбедио квалитет софтвера, може се увидети да је то сложен процес који захтева доста времена и преданог рада и пажње. Стога, на овом пољу је потребно још усавршавања и тражења бољих решења који ће на ефикаснији и поузданији начин да спроводе замисао квалитетног софтвера. Јер, испуњавањем овог циља, померају се границе у свету технологије и отварају се нове могућности које ће, поред ефикасности и лаког коришћења, красити и неопходан квалитет.

9. ЛИТЕРАТУРА

[1] Мирјана Бошковић, Олгица Николин, Улога и основне карактеристике квалитета софтвера, 2006.

[2] Виолета Томашевић, Развој апликативног софтвера, Универзитет Сингидунум, Београд, 2017.

[3] Миладин Стефановић, Зора Арсовски, Милан Матијевић, Обезбеђење квалитета софтвера, Машински факултет, Економски факултет, Крагујевац

[4] Which Software Quality Metrics Matter?

Веб сајтови:

[5] The Standish Group Reprot
<https://www.projectsmart.co.uk/white-papers/chaos-report.pdf> приступљено септембра 2020.

[6] Software quality
<https://asq.org/quality-resources/software-quality> приступљено октобра 2020.