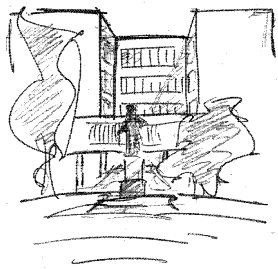


[P271]
Информациони системи

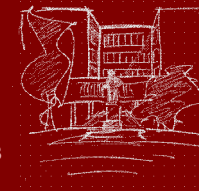
6



Саша Малков
Универзитет у Београду
Математички факултет
2023/2024

[P271]
Информациони системи

Саша Малков



Тема 10

Објектно оријентисане методологије

[P271] Информациони системи – Саша Малков – 2023/24 – час 6

1

Промене у односу на класичне методологије

Прилагођавање развојних методологија



- Неки од узрока проблема се могу отклонити измењеним приступом проблему развоја ИС
- Савремене развојне методологије почивају на концептима који омогућавају смањивање ризика

Универзитет у Београду – Математички факултет

[P271] Информациони системи – Саша Малков – 2023/24 – час 6

2

Промене у односу на класичне методологије

Савремени концепти развоја



- Неки од најважнијих савремених концепата развоја ИС који значајно смањују ризике су:
 - Инкрементални развој
 - Одређивање корака према роковима
 - Појачана комуникација међу субјектима
 - У жижи су објекти а не процеси
 - Прављење прототипова

Универзитет у Београду – Математички факултет

[P271] Информациони системи – Саша Малков – 2023/24 – час 6

3



Инкрементални развој

- **Принцип:**
 - Уместо да се развоју приступа као великом целовитом послу, са сложеним пројектом и имплементацијом, посао се дели у низ инкременталних фаза посвећених мањим целинама посла
- **Основна добит:**
 - Свака од фаза је значајно мања и једноставнија него читав посао, чиме се поједностављују планирање и имплементација
 - Већа је тачност планирања трошкова и рокова и тачније извештавање о току развоја
- **Основни ризик:**
 - Ако се у првим корацима потпуно занемаре предстојећи, постоји ризик да ће у наредним корацима бити потребне веће измене
 - Ако се у првим корацима узму у обзир сви предстојећи, постоји ризик да се њихова сложеност приближи сложености читавог система (“надувавање корака”), чиме овакав приступ губи смисао
 - Често није могуће сагледати унапред број, цену и укупно трајање свих корака



Одређивање корака према роковима

- **Принцип:**
 - За сваки инкрементални корак се прво одреде буџет и рокови, а тек после тога послови који ће кораком бити обухваћени
- **Основна добит:**
 - Дрastiчно смањивање ризика од прекорачивања буџета или рокова
 - Успостављање ритма редовног испоручивања нових верзија система
 - Редовнија контрола квалитета и виши степен поверења између субјеката
 - Постепено прилагођавање корисника новим елементима система
- **Основни ризик:**
 - Успостављање тесних оквира инкременталног корака може отежати поједине кораке и подићи укупну цену и трајање развоја
 - Неки елементи система се не могу природно поделити у различите кораке
 - Сваки корак захтева трошкове испоручивања што у збиру може да постане велика ставка у случају великог броја малих корака
 - У првим корацима се обично бирају послови који доносе већу добит. При крају развоја постоји ризик да се не имплементирају послови “чија је цена већа од добити” иако су значајни за систем као целину



Појачана комуникација међу субјектима ИС

- **Принцип:**
 - У планирању ИС (и сваког појединачног корака) укључују се у разматрање све врсте субјеката у што већем броју
- **Основна добит:**
 - Добија се тачнија слика о потребним циљевима из различитих углова
 - Смањује се ризик развоја неупотребљивог решења
 - Субјекти се кроз процес развоја припремају за употребу
- **Основни ризик:**
 - Превише информација може довести до претераног планирања, а тиме и до “надувавања” појединачних корака
 - Превеликим разматрањем мишљења субјеката који су навикли на постојеће процесе и не сагледавају планиране измене може се смањити обим суштинских функционалних измена у оквиру корака, а тиме и непотребно повећати број корака да се дође до “коначног” решења



У жижи су објекти а не процеси

- **Принцип:**
 - У средиште пажње се при развоју стављају објекти а не процеси
- **Основна добит:**
 - Објекти су обично стабилнији од процеса (тј. измене у начину пословања се мање одражавају на објекте него на процесе)
 - Такав приступ је прилагођенији инкременталном развоју
 - Смањује се ризик од погрешних одлука у раним фазама развоја
 - у раним корацима се више пажње посвећује објектима
 - у каснијим корацима се више пажње посвећује процесима
- **Основни ризик:**
 - Потпуно занемаривање процеса у раним фазама развоја може водити погрешној архитектури система, што се касније веома тешко (скупо) мења
 - Сасвим детаљно разматрање процеса у раним корацима прети да доведе до “надувавања” првих корака



Прављење прототипова

- Принцип:
 - У оквиру анализе проблема се прави прототип који одражава начин функционисања ИС
- Основна добит:
 - Олакшава се не-техничким субјектима да у раним фазама развоја уоче одређене недостатке
 - Смањује се ризик од погрешних одлука у раним фазама развоја
- Основни ризик:
 - Прототипови обично одражавају функционалне аспекте и елементе корисничког интерфејса, али не и унутрашњу структуру ИС. Последица је да су само делимично одговарајући ОО методологијама
 - Недовољно широко направљен прототип и nedовољно широко разматрање прототипа могу да прикрију недостатке у другим аспектима ИС
 - Превисше пажње посвећене прототипу прети да “надува” фазу његове израде



Околности развоја ООМ

- Постојање методологија које структурирано приступају анализи и описивању процеса
- Потреба за структурираним описивањем података
- Потреба за описивањем ентитета који мењају стања
- Подизање нивоа апстрактности посматрања елемената система
- Програмирање управљано догађајима
- Визуелни кориснички интерфејси
- Повећана модуларност софтвера
- Скраћивање развојног циклуса
- Транзиција модела
- Вишеструка употребљивост софтвера
- и друго



Објектно оријентисане методологије

- За разлику од структурних, које се у средиште пажње стављале уређивање процеса и алгоритама, ОО методологије у средиште пажње стављају уређивање објеката којима се описује систем
- Развој објектно оријентисаних методологија је почео у време када су слабости претходних методологија биле углавном познате
- У њих су уграђени неки од представљених савремених концепата развоја



Основни концепти ООМ

- Основни концепти Објектно оријентисаних методологија
 - У жижу стављају објекте, а не процесе
 - Све ООМ се одликују скраћивањем трајања развојних циклуса
 - РУП (и друге) прописују инкрементални развој
 - Агилне методологије прописују одређивање корака према роковима и трошковима
 - Појачана комуникација међу субјектима у свим фазама развоја



Основни концепти ООМ (2)

- Енкапсулација
- Интерфејс
- Полиморфизам
- Наслеђивање
 - Специјализација и генерализација
 - Хијерархије класа



Енкапсулација

- Структура објеката је њихова интерна ствар
 - не сме се излагати спољашњем свету
 - атрибутима се сме приступати само посредно
- Постоје случајеви када неки атрибути представљају основу понашања објеката или класа
 - тада им се може омогућити јавни приступ
 - и тада је ипак препоручљиво енкапсулирање
- Сврха
 - Апстраховање структуре методима
 - Виши ниво међусобне независности класе од модула у којима се употребљава



Интерфејс

- Објекат (класа) пружа спољашњим корисницима само скуп метода путем којих могу комуницирати са њим
- Такав скуп метода се назива *јавни интерфејс* објекта (класе)
- Интерфејс се обликује тако да омогући обављање једног целовитог посла
- Сврха
 - Суштина објеката је у њиховом понашању. Интерфејс омогућава то понашање.
 - Сужавањем интерфејса на суштину функционисања објекта прикрива се сва сложеност имплементације и пружа виши ниво независности објекта од окружења
- Напомена
 - Ако објекат има више интерфејса, значи да има више функција, па је потребно размотрити његово разлагање на више објеката



Полиморфизам

- Полиморфизам подразумева да се једном написан код може употребљавати за различите врсте објеката
- Постоје три основне врсте полиморфизма: хијерархијски, параметарски и имплицитни
 - Сви ОО програмски језици омогућавају хијерархијски полиморфизам
 - Неки савремени ООПЈ омогућавају параметарски полиморфизам
 - Само ретки језици подржавају имплицитни полиморфизам
- Сврха
 - Полиморфизам омогућава писање апстрактнијег кода, који има високу употребљивост

Наслеђивање

- Наслеђивање класе је еквивалентно увођењу једносмерне парцијално уређене релације “јесте” између класа
 - Класа А “јесте” класа Б ако сваки објекат класе А има све особине које имају и објекти класе Б
- Ако класа А “јесте” класа Б каже се и да је
 - А “изведена” класа из Б или А “је потомак” класе Б
 - Б “основна” класа за А или Б “је предак” класе А
- Релација “јесте” је парцијална релација поретка (рефлексивна и транзитивна)
 - Представља основу за грађење хијерархија класа
- Сврха:
 - Наслеђивање се користи за експлицитно означавање сличности међу класама (објектима)
 - Представља основу за хијерархијски полиморфизам: ако се нешто може урадити са објектом класе Б, онда се то може урадити и са сваким објектом класе А која је изведена из Б

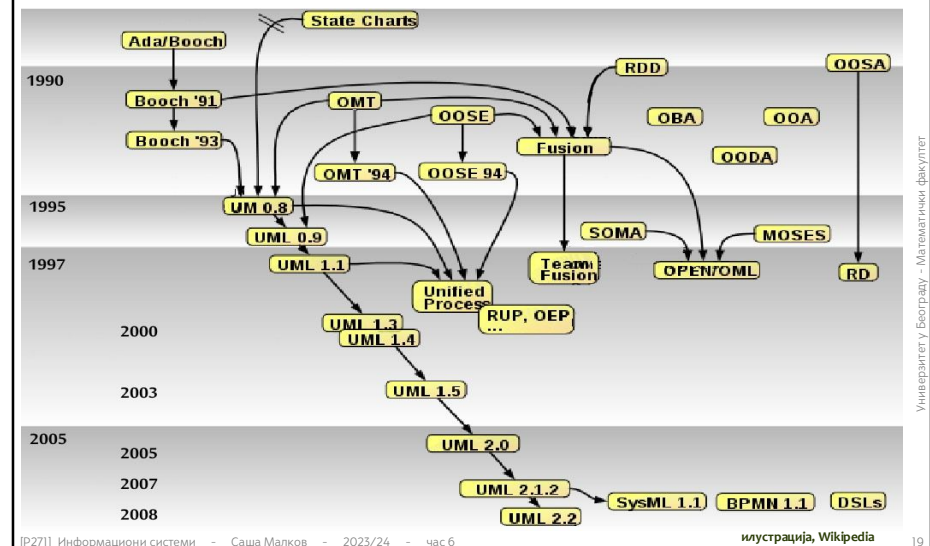
Специјализација и генерализација

- Наслеђивање се посматра у два смера, као
 - специјализација или
 - генерализација
- Ако класа А “јесте” Б, а класа Б “није” А онда:
 - класа А је посебан (специјалан) случај класе Б
 - класа Б је општији (генералан) случај класе А

Преглед ОО методологија

- Почетни кораци (-1997)
- Обликовање UML-а (1995-2005)
- Пост-UML кораци (2000-)

Преглед ОО методологија





Преглед ОО методологија (1)

- **Почетни кораци (-1997)**
 - Велики број различитих нотација и методологија
 - Ниједна комплетна и довољно широка
 - Акцент је на развоју метода и техника
 - *Booch, 1991.*
 - *Coad, Yourdon, 1991.*
 - *Martin, Odell, 1992.*
- **Обликовање UML-а (1995-2005)**
- **Пост-UML кораци (2000-)**



Преглед ОО методологија (2)

- **Почетни кораци (-1997)**
- **Обликовање UML-а (1995-2005)**
 - Неколико дубљих методологија концентрисаних на различите фазе развоја
 - Покушаји обједињавања метода и техника и уједначавања нотације
 - Акцент на нотацији
- **Пост-UML кораци (2000-)**



Преглед ОО методологија (3)

- **Почетни кораци (-1997)**
- **Обликовање UML-а (1995-2005)**
- **Пост-UML кораци (2000-)**
 - Уједначена нотација
 - Широко схватање процеса развоја
 - Потпуно посвећивање методологијама
 - РУП и друге савремене методологије



Основни концепти ОО методологија

- ООАД – Објектно оријентисани анализа и дизајн
- У центру пажње су објекти који фигуришу у систему:
 - ентитети
 - све врсте података који постоје у систему
 - субјекти
 - различите врсте корисника система
 - сервиси
 - услуге које систем пружа корисницима
 - интерфејси
 - подсистеми путем којих субјекти користе услуге



Основни елементи методологије

- ООАД се начелно не разликује много за ИС и мањи софтвер
- Основни поступци (*Coad & Yourdon, 1991.*)
 - Проналажење класа и објеката
 - Препознавање структура
 - Препознавање субјеката
 - Одређивање атрибута
 - Одређивање услуга



Проналажење класа и објеката

- Анализира се домен
- Препознају се стабилне класе и објекти
- Они ће представљати језгро будућег система и апликација



Проналажење класа и објеката (2)

- Посматрају се и траже:
 - структуре
 - други системи са којима посматрани систем сарађује или размењује информације
 - уређаји који се користе у систему
 - догађаји који се морају памтити или записивати
 - физичке и географска лоцираност, организационе јединице, састав тимова



Проналажење класа и објеката (3)

- Значајни аспекти при обликовању класа:
 - све што је потребно памтити чини потенцијалне атрибуте
 - потребно понашање чини потенцијалне методе
 - објекти обично имају више од једног атрибута, ако постоји само један, објекат се доводи у питање
 - класе обично имају више од једног објекта, ако постоји само један, класа се доводи у питање
 - атрибути морају да се односе на све објекте класе
 - понашање мора да се односи на све објекте класе
 - сви аспекти атрибута и понашања морају да потичу из домена проблема, а не из домена имплементације
 - избегавање редундантности и изведених атрибута



Препознавање структура

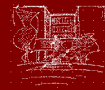
- На основу анализа класа и објеката утврђују се њихови међусобни односи
 - генерализација / специјализација
 - асоцијација
 - агрегација / композиција



Хијерархија

- Један од основних односа у ОО методологији чине односи генерализације / специјализације
- Они су основа за грађење хијерархија класа
- Основу за утврђивање ових односа представља понашање класа

- Поткласа је специјалан случај наткласе, за који важе сва правила понашања наткласе, али постоје и нека нова



Препознавање субјеката

- Циљ препознавања субјеката је у сегментацији система на подсистеме по критеријуму корисника
 - циљ има сличности са ДТП
- Посебно је значајно у великим системима са великим бројем услуга и / или класа података



Одређивање атрибута

- Одређивање атрибута класа и објеката се не разликује много од одговарајућег поступка у техници моделирања ентитета и односа
- Називи атрибута морају бити у складу са речником података
- За сваки атрибут се одређује тип и додаје опис



Одређивање услуга

- За разлику од структурних методологија, у ОО методологијама се пуна пажња услугама (поступцима, процесима) посвећује “на крају”
- Сервиси одговарају “атомичним” целинама послова, који би требало да се у пракси одвијају увек у потпуности
- Обично се препознају две врсте услуга:
 - рачунске услуге
 - услуге праћења и реаговања на догађаје
 - аутоматско
 - мануелно

Литература за тему



- *Avison, Fitzgerald, Information Systems Development (4.ed), McGraw Hill, 2005*
- *Simon Bennett, Steve McRobb, Ray Farmer, Object Oriented Systems Analysis and Design (Using UML), McGraw Hill, 2002*

Додатни извори

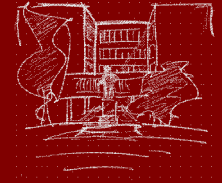


- *Booch, Object oriented design with applications (1.ed), Benjamin/Cummings, 1991.*
- *Coad, Yourdon, Object Oriented Analysis (2.ed), Prentice Hall, 1991.*
- *Martin, Odell, Object Oriented Information Engineering, Prentice Hall, 1992.*
- *Yourdon, Argila, Case Studies in Object-Oriented Analysis and Design, Prentice Hall, 1996*
- http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/
- <http://uml-tutorials.tyrem.com/>
- <http://www.uml.org/>

[P271]

Информациони системи

Саша Малков



Тема 11

Rational Unified Process

постулати, принципи, животни циклус



Rational Unified Process

- Основни елементи:
 - Скуп постулата и принципа на којима се заснива развој
 - Развојно окружење и скуп метода
 - Језик за описивање процеса и модела



Шта је RUP?

- Приступ развоју софтвера који почива на
 - итеративном циклусу
 - стављању архитектуре у средиште процеса и
 - управљању према случајевима употребе
- “Добро” дефинисан и структуриран процес развоја софтвера
- Методолошки производ са прилагодљивим окружењем



Зашто изучавамо RUP?

- Представља добар пример *велике* савремене методологије
- Објединио је већину значајних карактеристика старијих ОО развојних методологија
- И друге савремене методологије углавном почивају на истим тим карактеристикама
 - појмови, постулати, принципи, технике
- Флексибилан животни циклус



Основни елементи RUP

- Скуп постулата и принципа на којима се заснива развој
- Развојно окружење и скуп метода
- Језик за описивање процеса и модела



Појмови

- постулат (филозофија) – концептуална замисао и окосница на којој почива RUP
- принцип – правило или скуп правила чије поштовање се претпоставља у оквиру RUP-a
- посао (*task*) – врста активности коју обавља појединац или тим у оквиру RUP-a
- дисциплина – скуп сродних послова које може да обавља појединац или тим одговарајуће стручности
- резултат рада (артефакт) – појединачни резултат рада који настаје као последица обављања једног или више послова у оквиру методологије



Основни постулати

- Прилагођавање процеса
- Усклађивање сукобљених приоритета
- Сарадња целокупног развојног тима
- Итеративно представљање резултата
- Подизање нивоа апстракције
- Стално посвећивање квалитету



Основни постулати (2)

- Прилагођавање процеса
 - Концепти:
 - Сложеност процеса мора одговарати сложености пројекта и карактеристикама развојног тима
 - Формалност елемената процеса мора одговарати конкретним фазама
 - Добит:
 - Ефикасан развојни циклус
 - Повећана агилност пројекта
 - Реално планирање и компетентне процене



Основни постулати (3)

- Усклађивање сукобљених приоритета
 - Концепти:
 - Разумевање пословних приоритета
 - Одређивање приоритета пројекта и захтева
 - Упаривање потреба са могућностима софтвера
 - Разумевање којим се резултатима може управљати
 - Усклађивање поновне употребе резултата са потребама
 - Добит:
 - Усклађивање апликација са пословним и корисничким потребама
 - Смањивање заступљености специфичног развоја
 - Оптимизација пословне вредности



Основни постулати (4)

- Сарадња целокупног развојног тима
 - Концепти:
 - Мотивисање особља за пуно ангажовање
 - Прављење самосталних тимова
 - Подстицање мешовитих тимова (нпр. аналитичар, програмер, тестер...)
 - Обезбеђивање ефикасног окружења за сарадњу
 - Управљање производима рада (артефактима) у циљу подстицања сарадње
 - Интегрисање пословних, софтверских и извршних тимова
 - Добит:
 - Продуктивност тимова
 - Веће слагање пословних потреба и процеса развоја



Основни постулати (5)

- Итеративно представљање резултата
 - Концепти:
 - Омогућавање повратних информација инкременталним испоручивањем нових резултата корисницима
 - Прилагођавање планова развоја инкременталном процесу и добијеним одговорима корисника
 - Уважавање измена и управљање изменама
 - Суочавање са највећим техничким, пословним и оперативним ризицима у раним фазама
 - Добит:
 - Рано смањење ризика
 - Висока предвидивост
 - Поверење међу заинтересованим странама



Основни постулати (6)

- Подизање нивоа апстракције
 - Концепти:
 - Поновљена употреба постојећих резултата
 - Употреба алата и језика вишег нивоа, ради смањивања обима и сложености документације
 - Фокусирање на архитектуру
 - Пројековање са акцентом на квалитет, поузданост, разумевање и управљање сложеносту
 - Добит:
 - Продуктивност
 - Смањена сложеност



Основни постулати (7)

- Стално посвећивање квалитету
 - Концепти:
 - Пуна контрола тима над квалитетом
 - Рано и континуално тестирање, паралелно са развојем, интегрисањем и испоручивањем
 - Инкрементално грађење пројекта са аутоматизованим тестирањем
 - Добит:
 - Виши квалитет
 - Ранији увид у ток развоја и квалитет



Основни принципи процеса

- Визија
- План
- Ризици
- Пословни случајеви
- Архитектура
- Прототип
- Процењивање
- Измена захтева
- Подршка корисницима
- Прилагођавање процеса



Основни принципи процеса (2)

- Визија
 - Обликовање “визије” подразумева јасно сагледавање потреба и очекивања од пројекта
 - појмови
 - проблем
 - улагачи
 - карактеристике ИС
 - функционални захтеви (случајеви употребе)
 - нефункционални захтеви
 - друге претпоставке за остваривање пројекта
- Последице одсуства визије:
 - Промашена тема или циљ развоја, тако да резултат не задовољава очекивања



Основни принципи процеса (3)

- План
 - Прављење плана и управљања развојем пресудно утиче на резултате пројекта
 - планирање итеративног развоја
 - планирање састава и динамике ангажовања тимова
 - планирање протокола комуникације
 - планирање уједначавања документације
 - ...
 - Последица одсуства:
 - Слабо праћење тока развоја и недовољно и непоуздано остваривање осталих принципа



Основни принципи процеса (4)

- Ризици
 - Препознавање ризика и суочавање са најважнијим ризицима пресудно утичу на ритам и поузданост процеса
- Последица одсуства:
 - Фокусирање на погрешне проблеме
 - Могућност неочекиваних проблема



Основни принципи процеса (5)

- Пословни случај
 - Основна сврха “пословног случаја” је омогућавање економског и техничког планирања остваривања визије
 - Користи се за прецизно процењивање “профита” од пројекта
 - Представља оправдање за пројекат
 - Не бави се техничким карактеристикама проблема већ јасним и концизним образлагањем “зашто је пројекат потребан”
- Последица одсуства:
 - Ризик од бесциљног развоја, са непотребним утрошком времена и новца и ризиковањем одустајања од пројекта



Основни принципи процеса (6)

- Архитектура
 - Архитектура софтверског система представља организацију (структуру) најважнијих системских компоненти, њихових интеракција и интерфејса, са концептима уклапања у шире окружење.
- Последица одсуства:
 - Слаба могућност прилагођавања, скалирања, поновне употребе,...



Основни принципи процеса (7)

- Прототип
 - Прототип је средство за омогућавање улагачима без техничког образовања да остваре рани увид у будуће резултате пројекта
- Последица одсуства:
 - Одлагање уочавања проблема за касније фазе



Основни принципи процеса (8)

- Процењивање
 - Редовно сагледавање остварених резултата и процењивање њихове сагласности са планираном динамиком развоја је пресудно за успешно праћење и планирање
 - Свака итерација мора бити пропраћена анализом успешности и процењивањем квалитета и сагласности са плановима
- Последица одсуства:
 - Закаснило уочавање проблема и неисправно планирање итерација



Основни принципи процеса (9)

- **Измена захтева**
 - Измене захтева у току развоја су неминовна реалност и развој их мора уважавати и прихватати, али их и контролисати
 - свако представљање неког резултата, прототипа или итерације кориснику има за последицу талас измењених или нових захтева
 - формализовањем “захтева за промене” се уводи ред у овај иначе веома ризичан простор
- **Последица одсуства:**
 - Неисправно планирање итерација и постепено удаљавање од стварних потреба



Основни принципи процеса (10)

- **Подршка корисницима**
 - Сврха развоја је израда употребљивог производа. Употребљивост подразумева подршку корисницима.
 - упутство за употребу
 - подсистем за помоћ
 - кратка и јасна упутства на сваком од формулара
 - техничка подршка
 - обука
 - ...
- **Последица одсуства:**
 - Незадовољство корисника, слаб одзив и мањак квалитетних повратних информација, а на крају и неупотребљив софтвер



Основни принципи процеса (11)

- **Прилагођавање процеса**
 - Процес се мора прилагођавати променама у пројекту и окружењу у коме се пројекат развија
 - јаче управљање у случају слабих резултата
 - промене у начину одвијања процеса ради уштеде ресурса или подизања квалитета
 - ...
- **Последица одсуства:**
 - Лоша комуникација, неусклађеност тимова, неуједначеност документације, сувишан утрошак ресурса, низак ниво квалитета резултата...



Животни циклус пројекта

- Основу животног циклуса пројекта чине четири периода развоја:
 - Почетак (*Inception*)
 - Разрада (*Elaboration*)
 - Конструкција (*Construction*)
 - Прелазак (*Transition*)

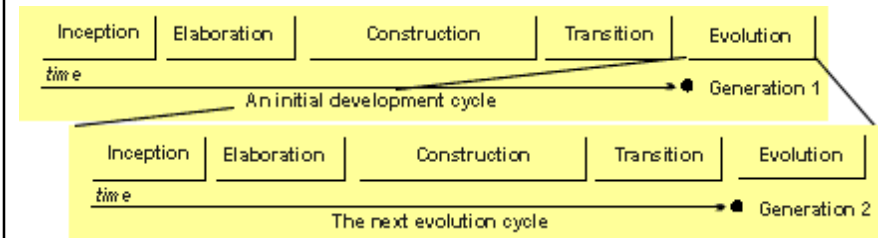


Животни циклус пројекта (2)

- Сваки од периода развоја
 - почиње по завршавању претходног
 - завршава јасно одређеним резултатом
 - на крају обухвата процењивање успешности и евентуално понављање неких поступака
 - може бити итеративан
- Читав развојни циклус може да се понавља
 - у истом или измењеном облику
- Појмови:
 - *развијни циклус* – један пролазак кроз четири периода
 - *генерација* – резултат једног развојног циклуса
 - *еволуциони циклуси* – развојни циклуси осим првог
 - обично су им краћи периоди Почетак и Разрада



Животни циклус пројекта (4)



Почетак (*Inception*)

- Одређивање *шта* се прави
 - (назива се и *Увод* или *Усвајање*)
 - визија
 - основни пословни случај
 - пројектни захтеви високог нивоа
 - иницијални случајеви употребе (10-20% довршени)
 - управљање оквирима пројекта



Разрада (*Elaboration*)

- Одређивање *како* се прави
 - детаљни захтеви (80% довршени)
 - дефинисање стабилне и извршне архитектуре
 - имплементирање неких кључних компоненти
 - око 10% кода имплементирано
 - заснивање архитектуре на кључним случајевима употребе
 - 20% случајева употребе одређује 80% архитектуре
 - пројектовање, имплементација и тестирање неких кључних случајева употребе



Конструкција (*Construction*)

- Прављење производа
 - довршавање затхева и пројектног модела
 - пројектовање, имплементација и тестирање сваке од компоненти
 - употреба прототипова и укључивање крајњих корисника у процес
 - инкрементална еволуција извршног система до готовог система
 - грађење дневних или недељних верзија са аутоматизованим процесом грађења
 - тестирање при сваком грађењу
 - испоручивање потпуно функционалног софтвера (бета)

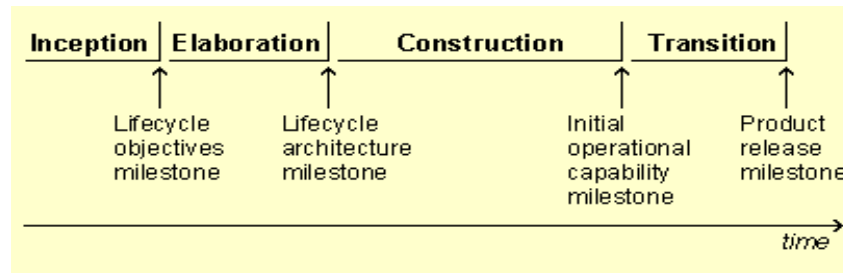


Прелазак (*Transition*)

- Испоручивање производа крајњим корисницима
 - прављење инкременталних издања ради отклањања неисправности
 - ажурирање корисничке и инсталационе документације
 - ажурирање описа издања
 - завршне анализе



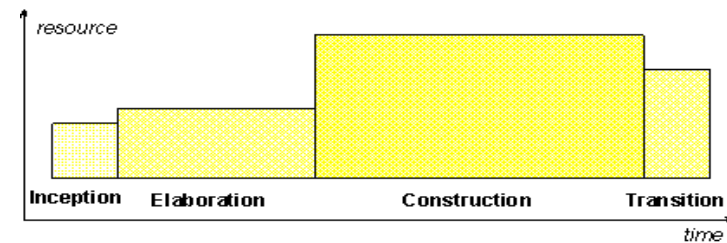
Животни циклус пројекта (3)



Животни циклус пројекта (5)

	Почетак	Разрада	Конструкција	Прелазак
Напор	5%	20%	65%	10%
Трајање	10%	30%	50%	10%

*ово су само илустративне пропорције – не морају да се стриктно поштују

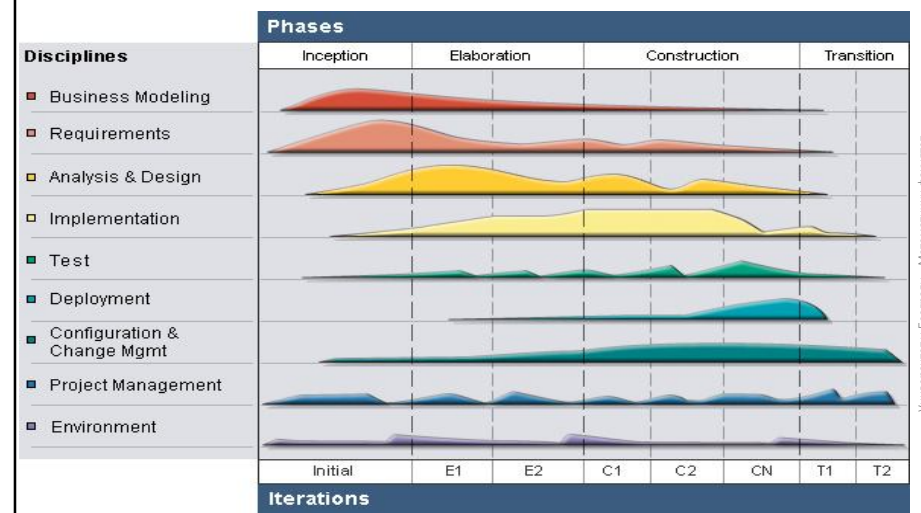




Животни циклус пројекта (6)

- Основне разлике у односу на приступ “водопада”
 - периоди развоја се не обликују према поступцима него према циљевима и очекиваним резултатима
 - сваки период развоја може да обухвати активности из различитих дисциплина, од прикупљања и анализе захтева, па све до имплементације и испоруке верзије
 - резултати претходних периода се могу мењати, унапређивати и дорађивати у наредним периодима развоја

Животни циклус пројекта (7)

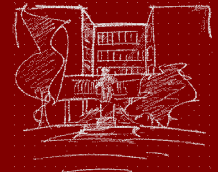


Литература за тему



- *Avison, Fitzgerald, Information Systems Development (3.ed), McGraw Hill, 2003*
- *Langer, Analysis and Design of Information Systems, 3.ed, Springer, 2008.*
- *Wong, Rational Unified Process Web Site, <http://alberttwong.com/wp-content/uploads/2008/11/RUP/>*

Хвала на пажњи!



МАТФ
Универзитет у Београду
Математички факултет

