

ЛЕКЦИЯ 1. Методи на информационното моделиране

Информационното моделиране е важна дейност при разработването на компютърно базирани информационни системи.

Информационният модел засяга част от организацията и нейното обкръжение и предоставя средства, с които информацията се записва и начини, по които информацията се обработва.

Информационните модели като цяло приемат формата на entity-relationship модели, диаграми за поток на данните (поточни диаграми) и история на entity.

Информационният модел се проверява от потребителите за да е сигурно, че техните изисквания за информационната система са разбрани коректно и пълно, и той се използва по-късно за проектиране на компонентите за информационна обработка от информационната система за тази организация.

Информационните модели на организацията често включват спецификация на изискванията. Стремекът към повторното използване на части от съществуващите системи, или реинженеринг на системата, е много труден без наличието на такива модели. Ако моделите съществуват в стандартна форма, комуникацията между потребителите и системните разработчици може да се подобри.

"Сближаването" засяга преместване от разнообразни, съществуващи методи към "един най-добър метод", докато „хармонизирането“ предполага съществуването на референтен метод, с който съществуващите методи имат явни съответствия.

Въпреки това, ако резултата от „сближаването“ е на достатъчно високо равнище то може да бъде еквивалентно на хармонизация.

Основни цели

Нашата първа цел е да анализираме и сравним възможностите на информационното моделиране за избраните методи за разработване на системи, за да разберем техните прилики и разлики, чрез референтната рамка от общи концепции и гледни точки за моделиране.

Втората ни цел е да покажем как такава референтна рамка може да се използва за метод на хармонизация чрез предоставяне въз основа за определяне на метод за съответствия.

Пет от методите, които сравняваме са информационния инженеринг (IE), структурирани системи за анализ и метод за проектиране (SSADM), Metodologica Informatica (MEIN), MERISE, и Coad и Yourdon's Обектноориентиран анализ (OOA).

Втора цел

Ще установяваме, дали методите имат широк или тесен обхват един спрямо друг и ще сравним обектноориентирания с традиционните методи.

Ще очертаем насоки за разработчика, където има избор на моделиране, за да бъдат направено и накратко посочено как направения избор може да се разглежда като предимства или недостатъци, свързани с набор от критерии за качество на информационното моделиране.

Ще обсъдим различните ситуации, които могат да благоприятстват или да не благоприятстват даден избор за моделиране. Такива насоки могат да бъдат част от една бъдеща, по-усъвършенствана референтна рамка.

В допълнение, за да подпомогнем разбирането на концепциите на метода и да покажем как те могат да бъдат използвани по интегриран начин, ще обсъдим общите принципи за изграждане на информационни модели, а също ще предоставим коментари и принципи, които описват как сме изградили специфични модели за примери от практиката.

Традиционни методи

Четири от петте метода, които ще сравним са представител на "традиционния" подход към развитие на системите. Този подход е в широка употреба през последните двадесет години и е основа за по-голямата част от софтуерни инструменти, които в момента на разположение за подкрепа на развитието на системата.

Три от методите, SSADM (Великобритания), MERISE (Франция) и Mein (Испания), са избрани, тъй като те са най-широко използваните методи в съответните европейски страни, докато четвъртия метод, софтуерен инженеринг, се използва широко в международен мащаб. В допълнение, тези методи са предмет на обсъждане в Euromethod, и са най-подробни по отношение на информация, моделиране, и са най-подходящи за нашето сравнение.

Обектноориентирани методи

Обектноориентирани методи имат предимства пред традиционните подходи. За да се проучат тези твърдения, петият метод с когото сравняваме е обектноориентиран анализ (ООА) (Coad и Yourdon 1991 г.), който е добре познат обектноориентиран метод за моделиране на информация.

Какво моделира информационния модел

Има различни позиции, между които моделиращия може да се колебае, когато решава какво ще се моделира, и позицията, която той ще заеме ще зависи от неговата гледна точка за естеството на действителността, което се нарича онтологичен поглед.

В контекста, с който сме свързани, гледната точка на наивния реализъм е, че организациите са обективни явления, които съществуват независимо и се възприемат по един и същ начин от всеки наблюдател, и модела за тази информация е модел на тази обективна реалност.

Към другия край на онтологичния спектър е това, което можем да наречем погледа на „субективизма“, който е че организацията е явление, чиято "реалност" е продукт на възприятията на наблюдателя, и следователно е индивидуална за всеки наблюдател. При този подход няма обективна реалност и информационния модел е модел на възприеманията на един или повече наблюдатели на организацията.

И двата възгледа са по-скоро крайни, и подхода който ще използваме е на социалния конструктивизъм. Той гледа на организацията като на социално изградено явление, чиито много аспекти, могат да се възприемат по различен начин от различните наблюдатели, но които са в състояние, чрез преговори, обсъждане или някакъв друг метод, да формулират общ поглед за период от време, съдържащ най-малко несъответствия, които могат да се използват като основа за изграждане на информационния модел.

Ще затворим това кратко описание на тази важна тема, като посочим важното предположение, което правим: процесът на постигане на обща гледна точка за организацията е завършен преди започването на дейността по моделиране и тази обща гледна точка е на разположение за тази дейност.

Какво е метод?

Метод за разработка на системи, или метод, може да бъде дефиниран, както следва:

Метод е интегриран набор от дейности и продукти за спецификацията и създаването на информационна система.

Процесът на прилагане на метод за разработване на информационна система се нарича *процес на развитие на системата*.

Определението се отнася до спецификацията или генерирането като някои методи не обхващат всички системи по процеса на развитие, както и произвеждането на спецификации, а не реалните системи.

Има няколко термина, които се употребяват. Те са до голяма степен еквивалентни като методологични методи, разработване, методи и подходи за разработване на системи. Техниката обикновено е по-специфична, като се позовава на добре дефинирана задача, която се занимава с определена област от процеса на разработване.

Фази и продукти

За да се намали сложността на метод, той обикновено се разделя на различни фази, състоящи се от дейности и по-малки задачи. Дейностите и фазите генерират продукти, които могат да бъдат, например - описанието на изискванията на потребителя, част от спецификацията, парче от програмен код или операционна система.

Концепция на модел, моделиране и моделиране

Един модел може да се дефинира общо като абстрактно представяне на част от реалността. Ще използваме термина модел, за да се позовем на абстрактно представяне на част от организация, която представлява продукт на частта за информационното моделиране на метод. Той е общ за тези модели, които включват аспекти на организационната среда.

Терминът моделиране се отнася до дейността на изграждане на модел, докато концепцията на моделирането е концептуален "градивен елемент", който е използван за изграждане на модел и това е осигурено от информационното моделиращата част на метод.

Трябва да отбележим, че терминът "модел" може да се използва по много различни начини в литературата за информационните системи.

Цели на методите

Като цяло, методите имат за цел създаването на информационни системи, които притежават високо качество и производителност.

Концепцията на високото качество е, че системата трябва да отговаря на изискванията на всички свои потребители, а на висока производителност концепцията е, че системата трябва да бъде разработена навреме и в рамките на бюджета.

Една некачествена система може да стане неизползваема или да бъде саботирана. Тя ще бъде загуба на ресурси за организацията и дори може да има ефект поставянето на организация извън бизнеса, особено ако засяга критична нейна част. Система с лоша производителност може да обезсили нейната полза, ако тя струва повече от планираното.

Еволюция на методите

Ранна ера (преди 1970)

В този период, са били идентифицирани само дейности по анализ и програмиране. Потребителските изисквания са анализирани в програмните спецификации от анализаторите на системи, които са ги насочили към програмистите. Дейностите са само слабо засегнати и само диаграма за прогреса на програмата е единственият стандарт.

Подчертаната дейност е била програмирането, като ефективността при изпълнението на програмата, по отношение на скоростта на изпълнение или на основния размер на паметта, са били основните съображения, поради хардуерните ограничения. Пример за метод от този тип във Великобритания е препоръчан от Националния компютърен център, описан от Daniels и Yeates (1969).

Структурирани методи - 1970 Структурно програмиране

Дейностите на фазата на програмирането постепенно става все по-определена, което отчасти се дължи на основен принцип, че само три конструкции - линейност, разклоненост и цикъл - са достатъчни за да се пишат програми с една входна и изходна точка (Bohm and Jacopini 1966; Dijkstra 1968).

Структуриран дизайн

При функционалния подход, Стивънс и др. (1974) предлага декомпозиция на система на определен брой йерархични функции или модули, показвайки ги на структурна диаграма и прилагайки правила за свързване и сближаване за управление на проектирането. Поточните диаграми са използвани за да се идентифицират основните трансформации на данни и подсистеми.

При данновия подход (Jackson 1975 г.; Warnier 1974), структурата на входните и изходните данни, е била използвана, за да се определи структурата на програмата по отношение на нейните йерархични функции.

Структурен анализ

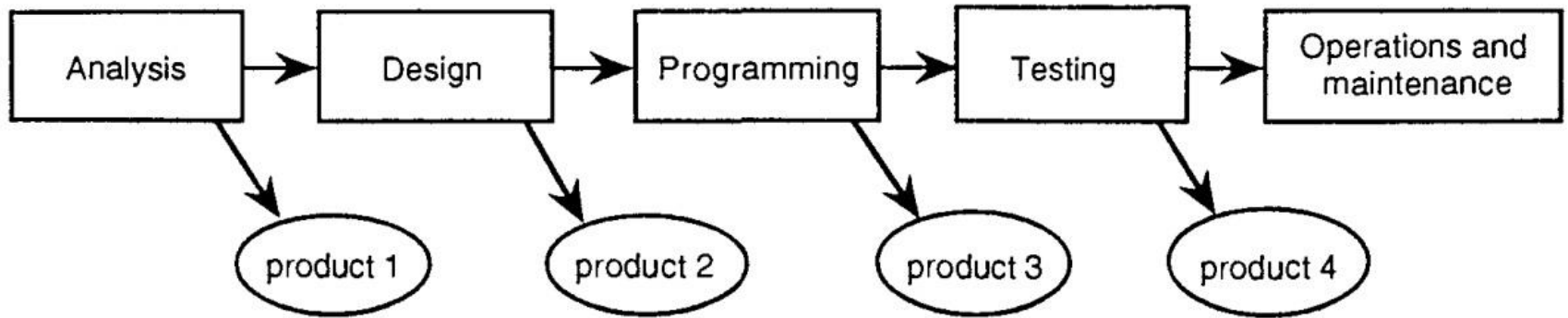
Той разграничава фазата на анализ от проектирането, с цел да осигури нетехническо описание на изискванията на потребителите. Той е съсредоточен върху диаграмата на потока от данни (DFD) (De Marko 1979; Gane and Sarson 1979; Yourdon and Constantine 1979), който моделира изискванията по отношение на йерархично разлагане на процеси и потоци от данни, вместо на програми и файлове на фазата на проектиране. Най-високо ниво на потока от данни на диаграмата дава преглед на системата за изясняване на изискванията. Осигурява улеснения за по-подробна спецификация, като например миниспецификации, таблици за решения, дървета за решения и речник на данните.

Модел „лавина“

Описани са тези фази и продукти от общ модел на развитие на системите, показан на фигура 1.

Моделът „лавина“ добавя все повече и повече детайли през следващите фази, създавайки все по-голям обем документация.

Разделението между фазите е важно, тъй като създава концепцията на абстракция. Ранните фази и техните продукти са на високо равнищна абстракция и се фокусират върху това какви са изискванията на потребителите, като се абстрахират от по-ниско ниво фази, които са загрижени как да бъдат изпълнени изискванията.



фиг. 1. Модел „лавина“

Ориентирани към данните методи - 1980

Структурните методи предполагат, че само потребителите искат да компютъризират текуща система, за да получат по-висока ефективност. Въпреки това, новите системи често изискват значително участие на потребителя и дискусия. Това води до акцент върху фазата на анализа.

В допълнение, структурните методи са процесоориентирани, с ефекта, че спецификациите често са твърде подробни и компютърно-ориентирани за потребителските разбираня.

Ориентация към данни

Това е моделиране на данни или обекти. То притежава ефекта на по-ясно разграничаване на етапа на анализ чрез разглеждане на изолирани от процеси обекти, което е по-абстрактно и окуражава нови начини да се мисли за организацията.

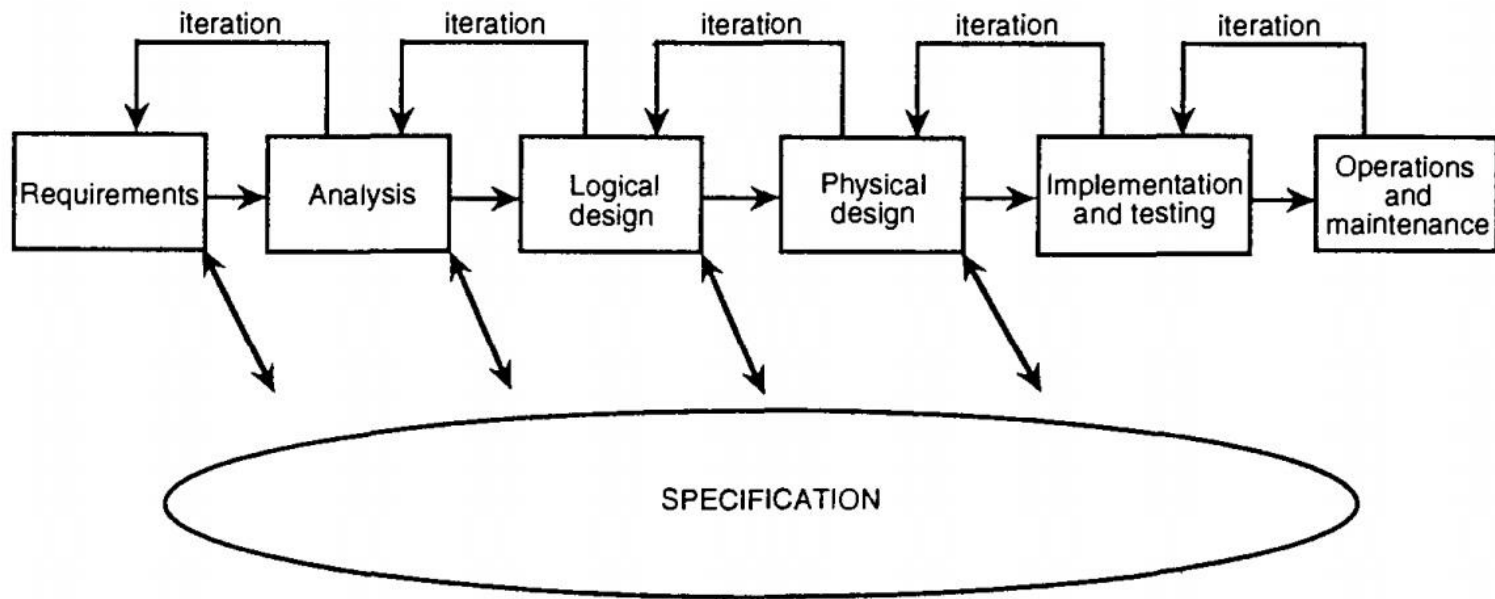
Ориентиране към потребителя ~ 1980

Съществувахе едно нарастващото разбиране, че много системи не отговарят на изискванията на потребителите. Необходимо е потребителите да се включат по-активно в процеса, така че те биха могли да проверят спецификацията. Следователно, продуктите са правени по-лесни за разбиране.

По-голям обхват от дейности за определяне на изискванията на потребителите се появи, като стратегическо планиране и определяне на разходите и ползите, които са полезни за по-новите видове системи, посочени по-горе.

Итеративен модел

Моделът „лавина“ се превърна в един итеративен модел на развитие на системите (фиг. 2). Итерацията е между една или повече фази, както и тя добавя акцент върху ранните фази, което води до появата на изисквания за определена фаза. Различните продукти, показани на фиг. 1, сега са част от една по-интегрирана спецификация.



фиг. 2. Итеративен модел за разработване на система

Итеративни фази на модел

Типичните дейности във всяка фаза могат да бъдат описани, както следва:

- *Изисквания.* Предизвиква и улавя изискванията от страна на потребителите.
- *Анализ.* Анализ на изискванията на потребителите в детайли и изразяването им точно в спецификацията.

Тази фаза основно се занимава с информационни дейности за моделиране, които изграждат информационни модели на важната част от организацията и околната среда, с акцент върху елементите, за които трябва да се запише информацията, и елементите, които изпращат и получават информацията и начина, по който информацията се обработва.

- *Логическа архитектура.* Спецификацията се използва за производство на проект за компютърно-базирана система, която ще служи като основа на внедрена информационна система. Често взаимодействието между човека и компютърната система се проектира тук.

Физически дизайн. Физическият дизайн е насочен към хардуера, софтуера, човешки и организационни компоненти.

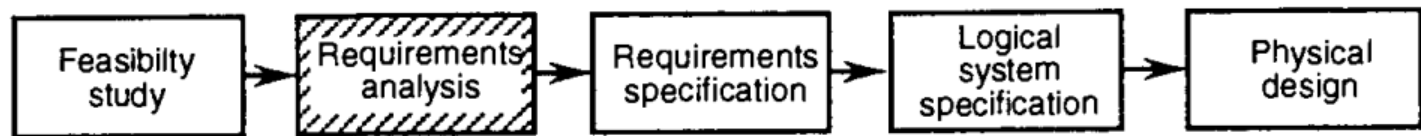
Внедряване и тестване. Системата е внедрена в софтуер и човешки процедури и е тествана.

Поддръжка. Поддръжката се състои от дейности, които искат и правят промени в системата.

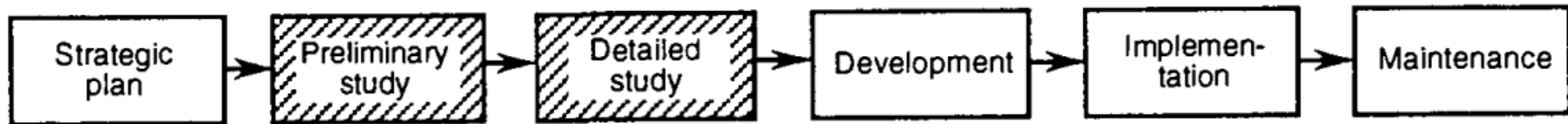
Традиционни методи

Четири от методите, които ще разгледаме са Информационен инженеринг, SSADM, MEIN и MERISE. Те са примери за методи ориентирани към данните и фиг. 3 показва техните основни фази.

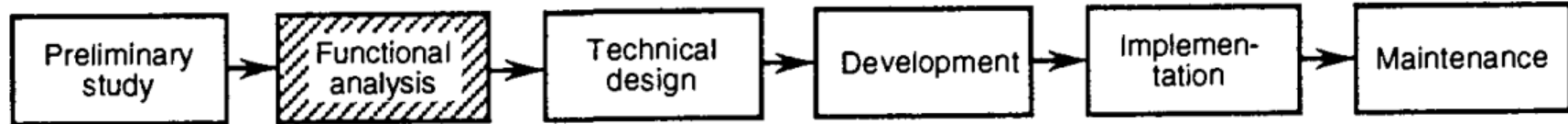
Интересуваме се предимно за информационните дейности за моделиране. Те се намират във фазата на анализ. Моделирането на информацията е важно, тъй като произвежда посочване на информацията и нейната обработка.



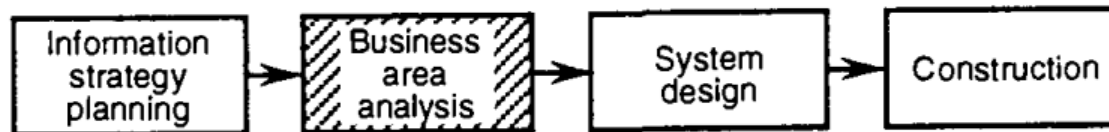
Structured Systems Analysis and Design (SSADM)



MERISE



Metodologica Informatica (MEIN)



Information Engineering (IE)

 Information modelling phase

Фиг. 3. Основни фази на Информационното инженерство, SSADM, MERISE и MEIN.

Всички методи имат предходна фаза, в която се определят изисквания, и последваща фаза, когато спецификацията се използва за логически дизайн.

Обектноориентирани методи

Ще сравним обектноориентирания (ОО) метод, обектноориентирания анализ (Coad и Yourdon 1991 г.) с четирите методи ориентирани към данни, които сме нарекли "традиционни" методи.

ОО подхода е еволюирал от първите ОО програмни езици, като например Simula 1967 и Smalltalk през 1970 г., с Object Pascal, Ada и C++, C# като по-съвременни примери за обектноориентирано програмиране; принципите за модулна спецификация (Парнас 1972) и работа на проектно равнище могат да се намерят по ниво проектиране в Jackson System Development (JSD) (Jackson 1983).

По-съвременните приложения на ОО принципите на проектиране и анализ на степента на развитие на системите е довело до обектноориентиран дизайн (OOD), както личи от Booch (1991) и обектноориентирания анализ, (OOA), които са представени в Coad и Yourdon (1991), Shlaer и Mellor (1988) и Rumbaugh et al. (1991). На архитектурата на базите данни е повлияла работата на Chen (1976) с връзки между обектите (ER). На транзакционният анализ е повлияла работата на Brodie и Silva (1982).

ОО методи навлизат в широка употреба в края на 80-те години на 20-ти век и обикновено са фокусирани върху отделните фази. Например, обектноориентиран анализ, обектноориентиран дизайн и обектноориентирано програмиране са фокусирани съответно върху фазите на анализа, дизайна и изпълнението.

Избираме Coad и Yourdon OOA за сравнение, тъй като той е на ниво анализ и се отнася главно за изграждане на информационен модел. Обектноориентираните методи са от особен интерес за нашето сравнение, тъй като те предлагат различна парадигма за информационно моделиране. Целта им е подобна на данновоориентираните методи, като техния фокус е върху обектите или данните, но те групират процеси по обекта, на който тези процеси работят.

Традиционните методи имат различни начини за групиране на процеси, и не интегрират процес с обект така силно.

ИНФОРМАЦИОННО МОДЕЛИРАНЕ

Информационното моделиране изгражда информационния модел, който обикновено се състои от две основни компоненти – структура на модела и процесите на модела.

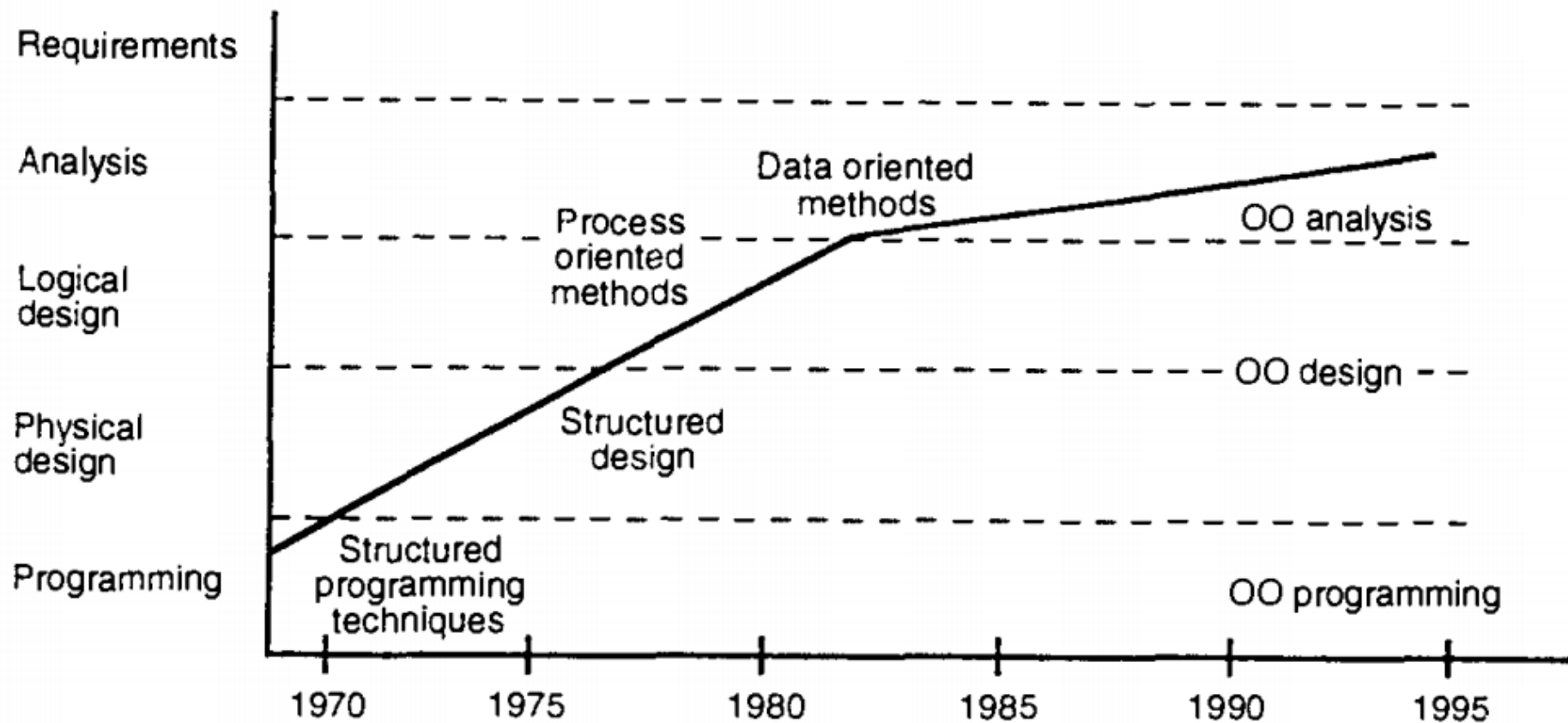
Структурата на модела описва организационните и заобикалящите организацията елементи, за които се налага да се записва информация, като често се използват концепциите за същност, атрибут и връзка, и се показват на релационнообектовата диаграма.

Процесите на модела описват елементите, свързани с обработката на информация, използвайки понятия като процес, събитие, поток данни и ги изразяват в термините на структурни моделни елементи. Например, процеси на модела са данновия поток или разделянето на процесите.

Възможно е да има трети компонент към информационния модел – „управление на модела“, което ограничава стойностите на елементите в структурата на модела.

Информационният модел е на концептуално ниво, така че данните, които засягат структурата на моделните обекти, атрибути и връзките няма да се получат преди да се премине проектната фаза на развитието на системата. На проектната фаза процесите ще бъдат проектирани да работят с тези данни, вместо с отделни елементи на структурата на модела.

Резюме на еволюцията на методите



Дали методите отговарят на целите си?

Методи за подобрения

Ще резюмираме начина, по който са се развили методите за постигане на високо качество и производителност.

- Фазите и дейностите, свързани с тях стават по-ясни, като заменят интуицията на разработчика за ориентиране и позволяват да се управлява по-добре процеса. Това намаля грешките на разработчика.
- Продуктите са определени по-ясно, така че изискванията на потребителите могат да се моделират правилно и точно във всички необходими подробности. Станали са по-прости, така че разработчиците да правят по-малко грешки и потребителите да могат да ги разберат и проверят.
- Ранните фази са станали по-абстрактни, улеснявайки включването на потребителите, което води до по-точното определение и улавяне на изискванията на потребителите, а също и помага за проверка на изискванията.
- Обектноориентираните методи обещава по-плавен преход между различните етапи, като през цялото време се използва обектноориентираната парадигма.

Ще разгледаме само онези проблеми, които се отнасят до по частта за информационното моделиране на методи, относно висококачествените цели на методите. Общите проблеми са, както следва:

Методите се различават по своя обхват и не могат да моделират всички аспекти на изискванията.

Съществува терминологичен проблем, тъй като методи могат да използват различна терминология за една и съща концепция и еднакви термини за различни концепции, които объркват разработчици и потребители.

Не съществува базов набор от концепции за моделиране, така че не е възможно да се осигурят добри насоки за разработчика относно дейности или продукти. Така лесно могат да се допуснат грешки при моделирането.

Моделите се изразяват във форма, която е трудна за разбиране от потребителите, правейки валидирането неефективно и водещо до ниско качество на системата.

По-нови методи и разглеждани проблеми

Ще разгледаме частта, която е свързана с информационното моделиране. Ще се концентрираме върху компютърно-базирания аспект на евентуалната система.

Широк критицизъм

Много критики и предложения за подобряване на метода са свързани с идентифициране на различни видове методи (Hirschheim и Klein, 1989) или подобряване на "твърди" методи, като ги променя в "меки" методи. Примери за по-широк вид критицизъм на методи са следните:

Методи предполагат изисквания, които са фиксирани и известни в началото на процеса на развитие. Въпреки това е добре известно, че изискванията се променят в процеса на разработка, поради фактори като околната среда или изискванията на потребителите, научили повече за технологията или тяхната организация.

Ако метод се съсредоточи върху грешен проблем в организацията, това би довело до осигуряване на система, която всъщност не е необходима, или не позволява достатъчно участие на потребителя, за изискванията, като за тях няма да бъдат определени правилно и напълно и валидирани спецификации.

Методи, които наблягат на техническите аспекти на информационните системи, пренебрегват по-човешкия социален и психологически контекст, в който те трябва да бъдат интегрирани.

Методите отнемат твърде много време, като разработчиците се съсредоточават върху сравнително маловажни задачи, като например подробно попълване на формуляри.

Насоченост на критиките

Като отговор на критиките, които посочихме по-горе, се появяват нови или подобрени методи.

Променящите се изисквания

Еволюционния подход (Crinnion 1991), е метод, който предполага разделяне на системата на няколко части, всяка от тези части е разработена отделно, така че потребителите имат възможност да научат постепенно техните изисквания. На по-общо равнище, Boehm (1985) е предложил спирален модел на разработване на системи, в които явно се включва итерация.

Грешен проблем / участие на потребителя

За намиране на ключови проблемни области в организацията могат да бъдат използвани фактори за критичен успех (Flin и Arce, 1995) или анализ на организационната цел (Yu, 1993 г.). Прототипиният подход (Ramesh и Luqi, 1993) предлага, като най-добро включване на потребители които да наблюдават експеримент с реалната система вместо абстрактни спецификации. Съществуват няколко метода, които включват софтуерни инструменти, които да направят прототипи на част или цялата система.

Само технически аспекти

„Меки“-те методи са насочени в тази област, и често обхващат широк набор от въпроси. Има методи, които предизвикват и договарят често сложни бизнес изисквания (Checkland, 1981), тъй като въвеждането на компютърните технологии изисква радикална преоценка на бизнес процеси (Hammer, 1990). Други методи имат за цел да предотвратят отрицателни социални и психологически ефекти, произтичащи от въвеждането на непозната или нежелана технология (Mumford, 1983 г.; Harker и др., 1990). По-конкретно, някои методи се съсредоточават върху "полезността" на евентуалната система за нейните потребители (Gould и др., 1991 г.).

Прекалено много време

Много методи се поддържат от CASE (Computer Aided Software Engineering) инструменти (Flynn и др., 1995), които управляват големи обеми от данни и автоматизират повтарящи се задачи в процеса на разработка.

СТАНДАРТИ

Тенденции

По-нова характеристика на развитието на метода е, че са предприети стъпки към стандартизация. Източниците за стандартизация се различават, и могат да бъдат индустриални, международни организации или органи, като Европейската комисия. В допълнение, обхватът на стандарта може да варира от най-високо равнище преглед на процеса на развитие на системите за подробно описание на различни видове продукти.

Много е вероятно стандартизацията да се увеличава в бъдеще, тъй като тя носи предимства, като например намаляване на разходите, опростено обучение и обещава по-висококачествени системи.

Съществуващи стандарти

Стандартът на Великобритания BS 5750 (Британски институт за стандарти, 1987) и неговия еквивалент, международния стандарт ISO 9001, е много общ стандарт, който обхваща управлението на всеки производствен процес, и много организации изискват от своите доставчици да се съобразят с този стандарт за производство на софтуер.

Схемата TickIT (TickIT 1992), първоначално спонсорирана от Министерството на търговията и индустрията във Великобритания, е процесът на сертифициране, който е по-специфичен за развитието на системите. Той оценява качеството на системата за управление на процеса на разработване на системи в една организация, като прилага стандарта ISO 9001-3, който е насочен към аспекти на управлението на софтуерното производство.

В САЩ, Capability Maturity Model (CMM), разработен под спонсорството на Министерството на отбраната (Humphrey, 1989) определя пет равнища на зрялост на процеса на развитие на системи в една организация. Всяко равнище на модела се характеризира с определени свойства, които се отнасят до процедурите, използвани за управление на процеса. Така например, организацията на равнище 1 притежава *ad hoc* или хаотичен процес на развитие, без формални процедури или механизъм за оценка на разходите.

Обзор на Euromethod

Текущото състояние на Euromethod (ССТА, 1994; Franckson, 1994) се занимава с определяне на резултатите, които да бъдат обменяни между клиента и доставчика по време на всички аспекти на развитие на системите, от офертата за възлагане на обществена поръчка до операционната система (Jenkins, 1992). Въпреки това, акцентът е по-скоро в началните, отколкото по-късните етапи на този процес.

Седем метода са избрани като основа за развитието на Euromethod: SSADM (Великобритания), MERISE (Франция), SDM (Холандия), DAFNE (Италия), Vorgehensmodell (Германия), MEIN (Испания) и Информационния инженеринг. Euromethod версия 0 е доставена през юни 1994 г. (ЕМ, 1994 г.), и след пробния период, версия 1 е от 1996 година. Първоначално тя се използва за рекламните обществени проекти за възлагане на обществени поръчки на Европейския съюз.

Отчетните резултати

Резултати се определят по отношение на общите понятия на модела на отчетните резултати, и се състои от три основни вида: целева област, план за доставка и резултати от проектната област.

Резултатите от целевата област са свързани с развитието на информационната система и се състоят от описание на информационната система и оперативните елементи. Планът за доставка описва отношения клиент-доставчик по време на процеса на разработка, дефинирайки началната, крайната и междинните стъпки на организацията и информационната система и последователността на точките, в които се взима решение. Проектната област разчита на процеса на разработка и се състои от проектните планове и отчетите по проекта.

IS-описания

IS-описание очертава елементите на информационна система и е еквивалент по смисъла на нашите означения на продукта. Тя може да се състои от един или повече от шест гледни точки, които в термините на IS-свойствата, описват различни, макар и припокриващи се множества от системни елементи. Заедно, гледните точки дефинират концепциите и терминологията описваща информационната система.

Три гледни точки са свързани с информационната система, а други три са свързани с компютърната система. Вижданията за информационната система определят представянето на информационния ресурс, използваните и създадените актьори и техните процеси с връзките между тях. Те са изгледи на: бизнес информация, бизнес процес и работна практика. Изгледите на компютърната система дефинират представянето на данните, структурата и функционалността на компютърната система. Те са погледа върху данните на компютърната система, нейните функции и архитектура.

Гледните точки за информационната система са на равнището на фазата на анализа, както вече описахме те дават представа за равнището на детайлност на бизнес процесите, което се състои от следните IS-описания: бизнес процес, стартиращи събития, генерирани събития, декомпозиция на процес, динамични зависимости, бизнес правила и използване на информация.

Хармонизация

Euromethod вижда две стъпки по пътя към хармонизиране на метода. Първо, той описва продуктите и концепциите на различни европейски методи по отношение на понятия и свойства на получавания модел, и на второ място, той предвижда ситуация, при която методи могат да доближават тяхната терминология към Euromethod.

ISO / IEC стандарт 12207

Този проект на стандарт (Singh, 1994) може да се разглежда като допълнение към Euromethod, тъй като не е документиран стандарт, но дефинира ключови процеси в развитието на системите. Това се прави така, че архитектурата е отворена и може да се използва с всеки модел на разработваната система.

Има три основни вида процеси: първични, поддържащи и организационни. Те покриват основните дейности по роли, като доставчик, получател, оператор, отговорник и ръководител на проекта. Архитектурата набляга на принципите на тотално управление на качеството и подчертава по-късните дейности в процеса на развитие, като дванадесет от четиринадесетте дейности са свързани с проектирането на системата или софтуер.

Бърза разработка на приложения - DSDM

Последният пример, който ще опишем ("радикална стъпка", Computing, 2 март 1995 г.) се нарича Метод за динамично развитие на системата (DSDM), и е стандарт за бързо разработване на приложения (RAD). Това е насочено към критиките, обсъдени по-горе, при която развитието на системите често е твърде бавен. DSDM осигурява на високо равнище рамка за разработване на системи, основана на тринадесет ключови принципи. Два от тези принципа са, че включване на потребителите в RAD е наложително, както и итеративно разработване и тестване, което трябва да бъде интегрирано през целия жизнен цикъл.