

Жизнен цикъл на информационните системи

1. Жизнен цикъл на софтуерните системи.

Разработването на софтуерните системи преминава през различни стадии и етапи, както бяха разгледани в лекцията на тема „Проектиране на информационни системи“. Жизненият цикъл (life cycle) обхваща всички фази (етапи) през които преминава създаването, използването, поддържането и снемането от експлоатация на системите. Ако фазите са свързани помежду си във времето говорим за динамичен цикъл, а за статичен, когато са ограничени във времето интервали с относително самостоятелни ресурси, за които могат да се определят начални и крайни състояния.

За класически жизнен цикъл се приема водопадния модел (waterfall model), който включва следните основни фази:

- 1) Анализ и дефиниция;
- 2) Проектиране
- 3) Разработка;
- 4) Тестване.

Във фаза 1) се определя приложната област на системите, клиентите и потребителите, дефинират се изискванията към системите. Резултати от тази фаза са: спецификация на изискванията и модел на системата.

Основната дейност във фаза 2) е свързана със създаването на детайлен проект на системата и определяне на софтуерната архитектура (спецификация на компонентите и техните връзки). Софтуерната архитектура представя процеса за изработка на софтуера, при които се избират: технологиите, които ще се използват; стандарт за писане на код; инструменти и платформа. Сложните задачи се разделят на по-прости и лесни за изпълнение. Разделят се компонентите и се описва на тяхната функционалност.

На фаза 3) се създава програмен код на системата, а на 4) се тестват отделните компоненти и тяхната интеграция в единна система. Тестовите са за неоткрити грешки, бизнес логика на софтуера и за сигурност. Резултати от последната фаза са тестовите протоколи и случаи.

Съществуват множество разширения на класическия жизнен цикъл, някои от които предлагат допълнителни фази или разделят фазите на части.

Според Денерт (Denert) всяка фаза се разделя на подфази. Например фаза 1) се разделя на следните подфази:

- Потребителски интерфейс;
- Функционален модел;
- Модел на данните.

Джакобсън (Jacobson) предлага разделяне на фаза 1) на две части: предварително проектиране и детайлно проектиране. За фаза 3) той предлага разделяне на три части: модулни тестове, интегрирани тестове и системни тестове. Така общия брой на фазите според Джакобсън е седем.

Балцерт (Balzert) предлага разширен модел на класическия жизнен цикъл в които фаза 1) се разделя на две подфазы: планиране и дефиниция. Фаза 4) не е самостоятелна, а е част от фаза 3). Добавя се нова фаза „съпровождане и въвеждане”.

През 1999г. Джакобсън, Буч и Рамбо предлагат унифициран процес на разработка на софтуерни системи, базиран на обектноориентирания подход и използването на UML (Unified Modeling Language). Според този процес фаза 1) се разделя на две отделни фази: изисквания и анализ. Във фазата „изисквания” се определят основните изисквания към системата. Те най-често се представят с диаграми на сценарии за използване (use cases). Във фазата „анализ” се определят основните класове и те се представят с диаграми на класовете.

Моделите на жизнения цикъл се разделят на две групи:

- ✓ Последователни (sequential life cycle models);
- ✓ Итеративни модели (nonsequential life cycle models).

При последователните модели фазите се изпълняват последователно. Всяка фаза започва след като е завършила предходната. Типични такива модели са:

- Водопадния модел (waterfall model);
- Фау-модел (V model);
- Cleanroom-Engineering

При итеративните (циклични) модели се допуска връщане към от една фаза в предишна. Такива модели са:

- Спирален модел (spiral model);
- Водовъртежен модел (whirlpool model);

- Фонтанен модел (fountain model).

2. Средства и подходи за разработване на софтуерни системи

В процеса на разработване на софтуерните системи, както и на информационните системи се използват различни средства като: различни видове диаграми, абстракции, псевдокод, специализирани езици и др. Прието е тези средства да се наричат **базови концепции**. Те се представят чрез **техники за описание (нотации)**. Например, бизнес процесите се представят с диаграми на сценарии за използване, BPMN диаграми, UML диаграми. Т.е. за някои базови концепции съществуват повече алтернативни нотации.

Основните задачи на базовите концепции са:

- да представят по-прецизно и еднозначно спецификацията на изискванията;
- коректно моделиране по време на целия жизнен цикъл.

Нотациите на базовите концепции се различават по вид и по степен на формализация. По вид те се разделят на: текстови (текстове на естествен език) и графични (различни графични символи). По степен на формализация нотациите се разделят на: формални (използват се формални езици или нотации); неформални; полуформални (комбинация от предходните).

Най-често използваните подходи за създаване на софтуерни системи са:

- ✓ Функционален;
- ✓ Обектноориентиран;
- ✓ Ориентиран към данни;
- ✓ Ориентиран към алгоритми;
- ✓ Ориентиран към правила;
- ✓ Ориентиран към състояния;
- ✓ Ориентиран към сценарии.

Подходите могат да използват една или повече базови концепции.

Например:

Функционалният подход включва следните базови концепции:

- Информационни потоци;
- Функционални йерархии;
- Бизнеспроцеси.

Ориентираният към данни подход използва:

- Структури от данни;

- Обектни типове и връзки между тях.

Обектноориентираният подход включва структури от класове и т.н.

Базовите концепции (средства) трябва да имат следните характеристики:

- ✓ Независимост – от фазите на жизнения цикъл; от приложенията за които се използват; методите за разработка; контекста;
- ✓ Автономност – завършеност, не може да бъде редуцирана в други базови концепции;
- ✓ Продължителност – по отношение на съществуване и използване.

По-подробна информация за връзката между подходи и средствата (базовите концепции) са представени в следващите подточки.

2.1. Базови средства на функционалния подход

При функционалния подход се използват основно функции и функционални дървета, диаграми на потока от данни и диаграми на сценарии на използване.

Функциите обикновено описват определени дейности и задачи в дадена приложна област. Йерархия от функции се представя с функционално дърво. Функциите и функционалните дървета са разгледани в лекцията на тема „Моделиране на процеси”, а диаграмите на потока от данни - лекцията на тема „Структурен и обектноориентиран анализ”.

Диаграмите на потока от данни (DFDs-Data Flow Diagrams) представят разработваната система като поток от данни (движение на информация) между функции, памет (хранилища за данни) и интерфейси към външни обекти. Данните могат да бъдат пренасяни или преобразувани от един в друг вид. Обикновено в тези диаграми се използват следните графични означения:

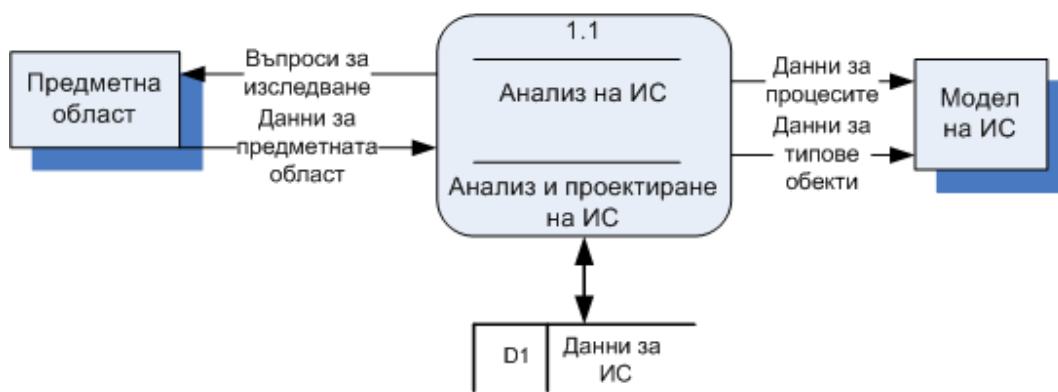
- ✓ именувана окръжност – за означение на функция;
- ✓ именувана стрелка – за поток от данни;
- ✓ именуван правоъгълник с името на интерфейса – за интерфейс към външни обекти;
- ✓ две паралелни линии, между които се задава името на мястото за съхранение на данни (паметта) .

При построяване на диаграми на потока от данни трябва да се спазват следните правила:

- Всяка диаграма трябва да съдържа поне един външен обект.

- Всеки външен обект се представя само веднъж.
- Всеки поток от данни има име. Без имена са само потоците които водят/идват от/към хранилищата.
- Между паметите не трябва да има директни потоци от данни.
- Между интерфейсите към външни обекти не се представят потоци от данни.
- Между интерфейсите към външни обекти и паметите не трябва да се задават директни потоци от данни.
- Между външните обекти и паметите трябва винаги да се задават функции.
- Диаграмите на потока от данни не съдържат възможности за разклонения и цикли.
- Интерфейсите трябва да дават ясна представа за оригиналния източник или предназначение на информацията.
- Добре е имената на потоците от данни да са съществителни имена или комбинация от прилагателни и съществителни.
- Имената на функциите е удачно да бъдат глаголи или съществителни, следвани от глаголи - например Регистриране на клиенти, данни за клиенти въвеждане и др.

На фиг.1 е представена една диаграма на процеса „Анализ на информационната системи”.



Фиг.1. Диаграма на процеса „Анализ на ИС” [<http://tuj.asenevtsi.com/APIS/APIS34.htm>]

Като предимства на диаграмите на потока от данни могат да се посочат:

- ✓ лесни за създаване;
- ✓ лесни за разбиране, дори и от непрофесионалисти;

- ✓ съдържат повече информация от функционалните дървета.

Като недостатъци на диаграмите на потока от данни могат да се отбележат:

- ✓ невинаги означаването на потоците от данни само с имена е достатъчно;
- ✓ трудно е да има единно абстрактно ниво за функциите и данните.

Диаграмите на сценарии на използване (use cases) описват работните потоци като бизнес процеси. Бизнес процесите се състоят от отделни дейности и могат да се представят със сценарии. Сценариите се представят като диаграми. Диаграмите описват бизнес процесите в информационната система, връзките между бизнес процесите, както и връзките между бизнес процеси и субектите (актьорите) на системата. Построяват се по следните правила:

1) Определят се субектите (актьорите) на ИС. Те могат да бъдат: един или група потребители; външни за ИС устройства или софтуер. Потребителите могат да заемат различни роли, докато актьорите представят клас (вид), на който се присвоява само една роля.

2) Идентифицират се сценариите по които актьорите взаимодействат със системата.

Сценариите могат да се представят с различни техники, като:

- Текстова схема (текстово описание) – шаблон за описание на естествен език;
- Диаграми на сътрудничество (collaboration diagrams) – представят сценариите като обекти от системата и връзки между тях. Отразяват и съобщенията обменяни между обектите;
- Диаграми на последователности (sequence diagrams) – акцентът при тях е последователното обменяне на съобщения във времето;
- Диаграми на действия (activity diagrams) – представят сценариите като алгоритми.
- Крайни автомати (state-transition diagrams) – представят поведението на обектите в системите и техните възможни състояния, заемани по време на изпълнение на сценария.
- Мрежи на Петри (Petri nets)– за представяне на паралелни процеси.

Компоненти на сценариите са: име (идентификатор); цел, която ще се постигне при успешно изпълнение на сценария; категория (основна,

второстепенна или незначителна); предусловия; успешни следусловия; неуспешни следусловия; актьори; описание; алтернативи.

2.2. Базови средства на ориентирания към данни подход

Ориентираният към данни подход обикновено се използва за проектиране на бази от данни (БД) и ИС с БД. Той използва като средства речник на данните и идентичност-връзка (ИВ) модел.

Речникът на данните (Data Dictionary) е каталог на данните. Той съдържа информация за техните свойства, структура и начин на използване. Съхранява структури от данни от потребителска гледна точка. За описание се използва нормална форма на Бакус –Наур (Backus-Naur-Form, BNF). Като нотация представя информацията от диаграмите на потока от данни или на класовете. Няма графично представяне.

Основните елементи на ИВ моделите, предложени от Чен (Chen) са: идентичности, асоциации (връзки), кардиналности, агрегации, роли. Идентичността е индивидуално представяне на обект (реален или абстрактен). Визуално графично се изобразява с правоъгълник. Идентичности с едни и същи характеристики могат да се групират в множества. Характеристиките се наричат атрибути. Атрибутите могат да бъдат:

- описателни – представят свойствата на идентичността;
- идентифициращи – използват се за еднозначна идентификация, например факултетен номер, ЕГН, клиентски номер и др.

Асоциациите отразяват връзките между множествата идентичности. Графично се визуализират чрез ромбове, в които се записват интерпретациите им.

Кардиналност на една връзка се изразява като брой на идентичностите от един тип, с които е свързана една идентичност от друг тип. Представя се със следните спецификации: 1 – при точно един свързан елемент; k-точно k елемента; С (Choice) – избор между нула и един елемент; М – избор между повече елементи (1,2,3...,n); МС – избор между нула, един и повече елементи (0,1,2,3,..., n).

Агрегацията е вид асоциация между множества от идентичности. Интерпретира се като връзка от типа едно множество е част от друго множество идентичности. Например идентичността „катедра” е част от

идентичността „факултет” ; идентичността „специалност” е част от идентичността „професионално направление” и др.

Ролите в ИВ модела се използват за означаване на функцията на идентичност в дадена асоциация. Например идентичността „катедра” – „обучаваща катедра” или „специализираща катедра”.

2.3. Базови средства на ориентирания към правила подход.

Основните средства използвани при ориентирания към правила подход са правила, таблици и дървета на решенията.

Правилата се представят обикновено със следния синтаксис:

If условие₁, **and** условие₂ ... **then** операция,

където условията описват ситуацията, необходима за да се извършат операциите. Операциите могат да бъдат:

- импликации или дедукции – логически стойности на определени твърдения;
- действия – за промяна на определени състояния.

Таблиците на решенията се състоят от четири квадранта (табл.1). В първия се задават условията. Във втория се записват действията. А в третия и четвъртия се използват за свързване на действията с условията. Свързванията се правят по колони. Всяка колона се интерпретира като правило и се свързва (идентифицира) с определен идентификатор.

Таблица 1. Структура на таблицата на решенията

	Име на таблицата	Номер на правило
If	1) Условие	3) Знак за условие
then	2) Действия	4) Знак за действие

Освен вертикално, таблиците на решенията могат да се представят и хоризонтално. За по нагледно представяне за потребителите могат да се използват дървета на решенията. При тях всички алтернативи се представят явно. Дърветата се обхождат от ляво на дясно.

2.4. Базови средства на ориентирания към състояния подход

При ориентирания към състояния подход най-често се използват крайни автомати, диаграми на действията и мрежи на Петри. Крайни автомати и диаграми на действията се използват за моделиране на

обектноориентирани софтуерни системи. За представянето им се използва езика UML.

Едни от най-често използвани крайни автомати са автомат на Мур (Moore) и автомат на Мили (Mealy). Теорията на крайните автомати е обект на друга дисциплина, която вече е изучена и затова няма да бъдат разглеждани в тази лекция. Чрез крайни автомати е удачно да се представят само обекти от класове, които имат динамично поведение. Те са удобно средство и за моделиране на жизнения цикъл на обектите. Крайните автомати могат да се представят с графи (графи на състоянията).

Мрежите на Петри са обект на дисциплината „Компютърно моделиране”, която предстои да бъде изучавана.

Диаграмите на действията описват алгоритмите и бизнес процесите като състояния на действията. Състоянията на действията са стъпките по време на изпълнение на един алгоритъм или бизнес процес. Едно състояние на действие е напуснато, когато дейността свързана с него е приключила.

2.5. Базови средства на ориентирания към сценарии подход

Базови средства на ориентирания към сценарии подход са съобщения, сценарии, диаграми на сътрудничество, диаграми на последователност.

Съобщенията се използват за моделиране на заявки на клиенти за изпълнение на услуги. Обикновено в диаграмите съобщенията, изпращани към обектите се представят с непрекъснати стрелки, а отговорите – с пунктирани стрелки. Последователността, в която обектите си разменят съобщения се представят чрез сценарии. Сценариите се представят с диаграми на взаимодействие. Те могат да бъдат диаграми на сътрудничество и диаграми на последователност.

Диаграмите на сътрудничество (collaboration diagrams) се използват за поэтапно моделиране на информационните потоци в един сценарий. Основните елементи на тези диаграми са обектите и разменяните между тях съобщения в даден информационен поток. Съобщенията се номерират за да могат да се управляват.

Диаграмите на сътрудничество могат да се използват и за моделиране на жизнения цикъл на обектите.

Диаграмите на последователност (sequence diagrams) показват съобщенията, които обектите си разменят последователно във времето;

сценарии за някои от процесите в една функционалност; обектите, които участват в тези сценарии.

В тези диаграми обектите се поставят най-горе. Не е необходимо номериране на съобщенията, защото те се изпращат така, както са подредени в диаграмата. Първо се изпраща съобщението, което е най-горе в диаграмата. Четенето на тези диаграми се извършва от горе на долу.

Диаграмите на последователност също могат да се използват и за моделиране на жизнения цикъл на обектите. В този случай за всеки обект се изобразява линията на живот. Най-често тя се визуализира с пунктирна линия и показва времевия интервал, в който обектът съществува. Изпращането на съобщение от един обект към друг се изобразява с насочена към обекта-получател стрелка. Обектите могат да изпращат съобщения и на себе си. Правоъгълниците върху линията на живот показват продължителността на действие на съответния обект. Това са интервалите от време, в които обектът е активен. Моментът на премахване на обектите обикновено се изобразява с кръстче върху линията на живота на обекта.