

18. Инженеринг на софтуерните изисквания. Техники за извличане, анализ и валидиране на софтуерните изисквания. Специфициране на изискванията. (АСИ)

Анотация: Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Цел, задачи на инженеринга на софтуерните изисквания.
2. Видове изисквания – класификация.
3. Същност на отделните етапи (дейности) на инженеринга на изискванията.
4. Техники за извличане, анализ и валидиране на изискванията, прилагани в отделните етапи на инженеринга на изискванията.
5. Начини за специфициране на изискванията. Видове модели в зависимост от перспективата на системата: модел на потока на данните, модели на поведението, семантични модели, обектно-ориентирани модели, формални модели.

Съдържание

1. Цели и задачи на инженеринга на софтуерните изисквания
 - 1.1. Какво правят инженерите по изискванията?
 - 1.2. Дейности на процеса ИИ
2. Видове изисквания – класификация
 - 2.1. Класификация според нивото на детайлност на описанието
 - 2.1.1. Бизнес изисквания
 - 2.1.2. Системни изисквания
 - 2.2. Хоризонтална класификация
 - 2.2.1. Функционални изисквания
 - 2.2.2. Нефункционални изисквания
 - 2.2.3. Изисквания, произтичащи от приложната област
3. Същност на отделните етапи на инженеринга на изискванията
 - 3.1. Извличане на изискванията
 - 3.2. Анализ на изискванията
 - 3.3. Документиране на изискванията
 - 3.4. Валидиране на изискванията
 - 3.5. Управление на изискванията
4. Техники за извличане, анализ и валидиране на изискванията, прилагани в отделните етапи на инженеринга на изискванията
 - 4.1. Техники за извличане на изискванията
 - 4.1.1. Сценарии, случаи на употреба
 - 4.1.2. Интервюта, въпросници, анкети
 - 4.1.3. Мозъчна атака
 - 4.1.4. Работни срещи
 - 4.1.5. Soft system methods
 - 4.1.6. Наблюдения и социален анализ
 - 4.1.7. Прототипиране
 - 4.2. Техники за анализ на изискванията
 - 4.2.1. Списък на изискванията
 - 4.2.2. Взаимодействия между изискванията
 - 4.2.3. Договаряне на изискванията
 - 4.3. Техники за валидиране на изискванията
 - 4.3.1. Преглед на изискванията
 - 4.3.2. Прототип
 - 4.3.3. Валидиране на модели
 - 4.3.4. Тестване на изискванията
5. Начини за специфициране на изискванията. Видове модели в зависимост от перспективата на системата
 - 5.1. Модел на потока на данните
 - 5.2. Модел на поведението
 - 5.3. Семантични модели
 - 5.4. Обектно-ориентирани модели
 - 5.5. Формални модели

1. Цели и задачи на инженеринга на софтуерните изисквания

Софтуерното изискване е характеристика, която системата трябва да притежава или е необходима на потребителя. То включва и документалното представяне на тази характеристика.

Процеса на инженеринг на изискванията е систематичен процес на идентифициране, анализиране, документиране и валидиране на функционалностите и ограниченията на даден софтуер, които са изисквани от клиента.

1.1. Какво правят инженерите по изискванията?

- Изявяват се в началото на проекта
 - Идентифицират проблеми, които се нуждаят от решения;
 - Анализът на изискванията е агент на промяната.
- Инженерите на изискванията трябва да:
 - Идентифицират потенциални проблеми;
 - Да са експерти в проблемната област (или да станат такива, ако не са).
- Взаимодействат с други субекти от организацията на процеса (управление на проекта, маркетинг и други)
- Фокусират се върху дейности по търсене и записване и класифициране на информацията

1.2. Дейности на процеса на ИИ

- Извличане на изискванията – чрез консултации със заинтересованите лица (ЗЛ).
- Анализ на изискванията и преговори на изискванията – преговорите служат за решаване на конфликти.
- Документиране на изискванията.
- Валидиране на изискванията – проверява се за съгласуваност и пълнота.
- Управление на изискванията – успореден с всички изброени дейности по-горе, управлява промените в изискванията.

2. Видове изисквания – класификация

2.1. Класификация според нивото на детайлност на описанието

- Изисквания на клиента и бизнес изисквания. Потребителски изисквания, според Съмървил.
- Системни изисквания.
- Спецификация на софтуерните изисквания.

2.1.1. Бизнес изисквания

Бизнес изискванията са целите на системата от високо ниво на организацията или клиента, които е поръчал системата. Те дефинират визията и обхвата на системата.

Има два вида бизнес изисквания – същински бизнес изисквания и изисквания за продукта.

Същински бизнес изисквания представляват резултати (стоки и услуги) от разработката на системата, които ще се предоставят след извършването на системата. Те обслужват бизнес целите чрез:

- решаване на проблеми;
- използване на възможности;
- посрещане на предизвикателства.

Потребителските изисквания описват целите на потребителите и задачите, които потребителите трябва да могат да извършат с продукта. Представят се чрез потребителски случаи, описания на сценарии и таблици със събития и отговори. Те доставят стойност само ако усовлетворяват същинските бизнес изисквания.

2.1.2. Системни изисквания

- Детайлно описание на услугите и ограниченията, което се характеризира с пълнота и консистентност;
- Обикновено съдържа структурни модели;

- Описва изисквания за продукта като система, тоест описва съществуващите зависимости и връзки;
- Основа за сключване на договор.

2.2. Хоризонтална класификация

- Функционални изисквания;
- Нефункционални изисквания (качествени изисквания + ограничения);
- Изисквания, произлизащи от приложната област.

2.2.1. Функционални изисквания

Функционалните изисквания (ФИ) описват какви функционалности трябва да предоставя системата на клиента или други системи и необходимото поведение от гледна точка на нужните дейности. В някои случаи указват какво не трябва да прави системата.

Те се документират чрез три допълващи се, но частично припокриващи се перспективи:

- Перспектива на данните;
- Функционална перспектива;
- Перспектива на поведението.

Описанието не трябва да бъде твърде общо, за да се избегне неяснота, но и не трябва да бъде твърде специфично, за да се избегнат твърде обемни документи.

2.2.2. Нефункционални изисквания

Нефункционалните изисквания (НФИ) са тези изисквания, които не се отнасят до функционалността на системата. Те поставят ограничения върху софтуерния продукт и процеса на разработката му. Уточняват външните ограничения, с които трябва да е съобразен продукта. НФИ се отнасят до системата като цяло, а не до някоя отделна функционалност. Състоят се от качествени изисквания и ограничения (организационни изисквания).

- **Качествени изисквания**

Качествените изисквания трябва да описват някаква качествена характеристика, която трябва системата да притежава като поведение. Например: кратко време за отговор, висока надеждност и т.н.

Видове качествени изисквания:

- Наличност – процент от времето, през което системата е налична и работи изцяло;
- Ефективност – мярка за това колко добре системата използва хардуерните си ресурси;
- Гъвкавост – колко усилия са необходими да се разшири системата с нови възможности;
- Цялостност – колко добре е защитена срещу непозволен достъп, нарушаване поверителността на данните, загуба на информация и заразяване с вреден софтуер;
- Съвместимост – колко лесно системата може да обменя данни с други системи;
- Надеждност – способността една система да продължи да функционира, когато трябва да се справи с определени входни данни, нарушения в свързаните системи или непредвидени условия за работа;
- Използваемост – усилията, изисквани от потребителите за да подготвят входни данни, да използват системата и да интерпретират изходните данни;
- Възможност за поддръжка – доколко е лесно да се поправи недостатък или да се направи промяна в системата;
- Преносимост – усилията за извършване на миграция на системата или компонент от една ОС в друга;
- Възможност за повторно използване – доколко един компонент може да се ползва в система, за която не е бил разработен първоначално;
- Възможност за тестване – степента на леснота, с която софтуерните компоненти или цялата система могат да се тестват за откриване на дефекти или недостатъци.

- **Организационни изисквания**

Ограниченията представляват организационно, технологично или нормативно изискване, което определя начина, по който ще бъде разработена системата.

Организационните изисквания са изисквания, които са ограничения върху процеса на разработка, зависещи от съществуващи политики и установени практики в организациите на разработчиците и на клиентите. Те биват три вида:

- Методи и стандарти, които трябва да се следват;
- Изисквания за имплементация – използван език за програмиране, метод за проектиране, технология за разработка и т.н.;
- Изисквания по доставянето – срокове и отчети, които трябва да бъдат доставени.

• **Външни изисквания**

Външните изисквания са резултат на външни за системата и за процеса на създаването ѝ фактори. Те могат да са наложени както върху продукта, така и върху процеса на разработване. Това са неща като необходимостта да работи с други системи, съобразяване с правила, закони и корпоративни политики.

2.2.3. Изисквания, произтичащи от приложната област

Изискванията, произтичащи от приложната област (ИППО) произлизат като резултат от областта на приложение, а не от изискванията на клиента. Тези изисквания могат да бъдат ФИ и НФИ и да произлизат от основни природни закони.

3. Същност на отделните етапи на инженеринга на изискванията

3.1. Извличане на изискванията

Извличане на изискванията е процес на търсене, откриване и обобщаване на софтуерните изисквания от наличните източници. Включва работа с клиентите, за да може да се проучи приложната област и внимателно да се анализират изискванията на ЗЛ.

Дейности по идентифицирането на изискванията:

- Разбиране на приложната област – знания за областта, в която се прилага системата;
- Разбиране на проблема – детайлите за специфичния клиентски проблем;
- Разбиране на бизнеса – как си взаимодействат системите и как спомагат за общите изисквания;
- Разбиране на нуждите и ограниченията на ЗЛ.

3.2. Анализ на изискванията

Целта на анализа на изискванията е да открият проблеми като незавършеност или несъгласуваност в идентифицираните изисквания, както и да се обсъдят и разрешат със ЗЛ в процеса на преговорите. Той е зависим от опита и експертизата на хората, които са включени в процеса на анализа.

Какво проверява анализа:

- Проверка за необходимост – анализира се необходимостта на изискването, за да не се допуска изискване, което не допринася за постигането на бизнес целите;
- Проверка за съгласуваност и завършеност – прави се cross-check проверка;
- Проверка за осъществимост – изискванията се проверяват за това дали са изпълними в контекста на бюджета и плана за разработването на системата.

3.3. Документиране на изискванията

Специфицирането на изискванията е процесът на записване на потребителските и системните изисквания в документ на изискванията.

- Изискванията на потребителите трябва да бъдат разбираеми от крайните потребители и клиентите, които нямат техническа подготовка;
- Системните изисквания са по-подробни изисквания и могат да включват повече техническа информация;
- Системните изисквания могат да бъдат част от договор за разработване на системата.

Независимо от използвания език за програмиране и метод на работа, три аспекта трябва да бъдат документирани за функционалността:

- Данни – как се използват и как са структурирани;
- Функции – условия за ползване, начин на обработка на входните данни и очаквани резултати;
- Поведение – следи се динамиката на проявление (синоним на инициатива).

Отделно имаме допълнителни аспекти за качеството и за ограниченията.

3.4. Валидиране на изискванията

Валидирането на изискванията е процес на проверка на това дали изискванията дефинират системата точно така, както клиента го иска. Това е важен процес, защото поправка на изискванията след доставянето на софтуера би струвало стотици пъти повече от поправка на имплементационна грешка по време на разработка.

Ако анализът работи със сурови изисквания, директно извлечени от ЗЛ, то валидирането работи с крайния чернови вариант на формулираните изисквания (тоест, те са вече договаряни и съгласувани).

Същност на валидирането:

- Удовлетворява, че документът, описващ изискванията е приемливо описание на системата;
- Документът на изискванията се проверява за пълнота и последователност, конфликти, двусмислици, съответствие със стандартите, грешки в моделите и технически грешки.

3.5. Управление на изискванията

Управлението на изискванията е процес на управление на промените на изискванията на системата. Главните задачи на управлението на изискванията са:

- управление на промените в уговорените изисквания;
- управление на връзките между изискванията;
- управление на зависимостите между документа на изискванията и други документи, създадени в процеса на инженеринг на изискванията.

Изискванията не могат да бъдат управлявани ефективно без да има проследимост на изискванията.

4. Техники за извличане, анализ и валидиране на изискванията, прилагани в отделните етапи на инженеринга на изискванията.

4.1. Техники за извличане на изискванията

4.1.1. Сценарии, случаи на употреба

Сценариите са примери от реалния живот за това как може да се използва системата. По-подробно, те са точно определена последователност на взаимодействие между актьора и системата.

Предимства: много са естествени и обикновено ЗЛ ги ползват спонтанно. Кратките сценарии са много добри за бързо илюстриране на специфични взаимодействия. Недостатъци: липса на структура, трудно се проверяват за пълнота.

Потребителските случаи са техника, базирана на сценариите, която идентифицира актьорите в едно взаимодействие и която описва самото взаимодействие. Наборът от потребителски случаи описва всички възможни взаимодействия със системата.

4.1.2. Интервюта, въпросници, анкети

Изискванията се обсъждат с различни заинтересовани лица, за да се изгради разбиране за техните изисквания. Интервюиращите трябва да бъдат непредубедени и да не провеждат интервюта с предварително формирани идеи за това какво се изисква. Интервютата се фокусират върху въпроси за:

- Бизнес случай за проекта;
- Функционални изисквания за проекта;
- Рискове;
- Ограничения;
- Заинтересовани лица.

4.1.3. Мозъчна атака

Също позната като brainstorming, това е тактика, която представлява работа в екип и предизвиква откриването на знания от всеки участник, което помага при разглеждането на алтернативи и правене на правилни избори.

4.1.4. Работни срещи

Анализаторът обединява основните ЗЛ, за да анализира системата и да разработи решението. Ръководителят на срещата поддържа фокусирана дискусия, протоколчик документира дискусията.

4.1.5. Soft systems methods

Това не са техники за детайлно идентифициране на изискванията. Те са по-скоро начин да се разбере даден проблем и неговия контекст в организацията.

Етапи на SSM:

- Оценка на проблемната ситуация;
- Описание на проблемната ситуация;
- Абстрактна дефиниция на системата от избрани гледни точки;
- Концептуално моделиране на системата;
- Сравнение модел/реален свят;
- Идентификация на промяната;
- Препоръки за действие.

4.1.6. Наблюдения и социален анализ

Етнографът прекарва известно време в наблюдение на хората по време на работа и така си изгражда представа за това как се извършват дадените процеси. Етнографията може да се използва за извличане на социални изисквания и някои организационни такива. Тя е хубаво да се комбинира с други техники, като например интервюта.

4.1.7. Прототипиране

Прототипа е първоначална орязана версия на системата, която може да се използва за експериментиране и презентация. Експериментирането подпомага изясняването и допълването на изискванията. Прототипите могат да прерастнат в крайни продукти и да изследват проектни решения и алтернативи.

Особености:

- Бързото разработване на прототипа е от висока важност, за да може да бъде наличен в ранния етап на процеса на идентификация;
- Прототипа може да се използва както при извличането, така и при валидацията на изискванията.

Подходи за бързо прототипиране:

- На хартия – създава се хартиен мокъп и се използва за системни експерименти;
- Wizard of Oz прототипиране – отговорите на системата за определени входни данни се симулират от човек;
- Автоматизирано прототипиране – използва се език от 4-то поколение или друга среда за бързо разработване, за да се създаде прототип (Бележка: 1-во поколение – машинен код; 2-ро поколение – асембли; 3-то поколение – C, C++, Java; 4-то поколение – Unix Shell, SQL, MATLAB, R).

4.2. Техники за анализ на изискванията

4.2.1. Списък на изискванията

Списък от ключови и достатъчно общи въпроси за оценка на всяко едно изискване.

Един примерен такъв списък:

- Преждевременно проектиране – включва ли изискването предварителна информация за проектирането или реализацията.
- Обединени изисквания – може ли едно изискване да се раздели на няколко?
- Ненужни изисквания – дали изискването е козметична добавка и в действителност не е необходимо.
- Използване на нестандартни компоненти – изискването предполага ли използването на нестандартен хардуер или софтуер?
- Следване на бизнес целите – дали изискването е в съгласие с бизнес целите.

- Неяснота на изискванията – могат ли да се прочетат по различен начин от различни хора?
- Реалистичност на изискванията – изискването реалистично ли е при технологията която използваме?
- Възможност за тестване на изискванията – може ли да се тества от QA инженерите?

4.2.2. Взаимодействия между изискванията

Много важна цел на анализа на изискванията е да се открият взаимодействия между изискванията и да се посочат конфликтите и припокритията между тях. Това се прави чрез матрица на зависимостите и взаимодействията.

4.2.3. Договаряне на изискванията

Договарянето на изискванията е процес на обсъждане на конфликтите между изискванията и постигане на споразумение, с което всички ЗЛ са съгласни. Разногласията за изискванията са неизбежни при наличието на много ЗЛ. При планирането на процеса на ИИ е важно да се остави достатъчно време за преговори.

Стъпки:

- Обсъждане на изискванията – проблемните изисквания се обсъждат, участващите ЗЛ представят своите възгледи за изискванията.
- Приоритизиране на изискванията – спорните изисквания се приоритизират, за да се идентифицират критичните изисквания и да се подпомогне процеса по вземане на решение.
- Споразумение на изискванията – откриват се решения на проблемите и се уговаря компромисен набор от изисквания.

4.3. Техники за валидиране на изискванията

- Преглед на изискванията – систематичен ръчен анализ на изискванията.
- Прототип – използване на изпълним модел на системата за проверка на изискванията.
- Валидиране на модели – Проверка за точността на моделите, представени в спецификацията.
- Генериране на тестови случаи – разработване на тестове за изискванията.

4.3.1. Преглед на изискванията

Най-често използваната техника за валидиране на изискванията – формулира се специална група, която да се:

- запознае с изискванията и да ги анализира;
- формулира забелязани проблеми;
- дискутира възможни действия за справяне с проблемите.

Разходът за прегледа на изискванията би могъл да бъде намален ако се направи предварителна проверка от един човек. Целта е да се минат най-явните пропуски, и когато се дадат на повече хора, да не се налага всички да поправят същите очевидни грешки.

Дейности на прегледа:

- Планиране – определя се екип, ръководител и протоколчик. Избира се време и място за срещите.
- Разпространение на документите до участниците в екипа.
- Подготовка за прегледа – всеки участник прочита документа и набелязва открити конфликти, пропуски, отклонения от стандартите и други.
- Среща по прегледа – дискутират се коментари и проблеми от отделни участници на екипа и се дефинира набора от възможности за справяне с проблемите.
- Проследяване на действията – ръководителят на екипа проверява дали формулираните действия са изпълнени.
- Ревизиране на документа – документа се редактира за да отрази съгласуваните действия.

4.3.2. Прототип

Тази техника се прилага само ако системата има вече разработен прототип. В противен случай времето и средствата загубени в този прототип няма да си струват.

Прототипите имат за цел да демонстрират изискванията и да помогнат на ЗЛ да открият проблеми. Те трябва да бъдат достатъчно пълни, ефикасни и robust, за да се оцени използването на исканата система. Валидирането може да започне с непълен прототип, но задължително системата трябва да бъде допълнена и довършена по време на валидирането.

Действия при прототипизирането:

- Избор на подходящ екип за тестване на прототипа – потребители с умерен опит и които са „отворени“ към използването на нови системи.
- Разработване на тестови сценарии – при внимателното планиране се получава формулиране на достатъчно пълен набор от тестови сценарии.
- Изпълнение на сценариите – хубаво е потребителите да ги изпълняват самостоятелно – по този начин стимулират реалната ситуация на използването на системата.
- Документиране на проблемите – съставяне на електронен или хартиен документ с откритите проблеми.

4.3.3. Валидиране на модели

Цел на валидирането:

- Проверява дали моделът е логичен, съдържа цялата необходима информация и няма конфликти.
- Проверява дали моделите за една система са вътрешно и външно съвместими.
- Проверява дали всеки модел представя реалните изисквания на клиента.

Някои проверки са възможни с автоматични софтуерни инструменти – CASE.

4.3.4. Тестване на изискванията

За всяко изискване трябва да може да бъдат съставени тестове за проверка дали изискването е добре дефинирано. Невъзможността да се създаде тест означава, че има липсваща или неясна информация в описанието на изискванията. Всяко функционално изискване в документа трябва да бъде анализирано с подходящ тест.

Няколко въпроса, подпомагащи съставянето на тест за изискванията:

- Каква употреба ще се тества?
- Съдържа ли изискването достатъчно информация, за да позволи разработването на тест?
- Нужен ли е повече от един тест за правене на изследването?
- Може ли изискването да бъде преформулирано, за да бъдат тестовите случаи по-ясно дефинирани?

Формуляр на тест за изискванията.

- Идентификатор на изискването.
- Свързани изисквания.
- Описание на теста – да е кратко, да обясни защо теста е обективен за даденото изискване, описание на входно-изходната информация.
- Проблеми – описание на проблемите, които правят теста труден или невъзможен.
- Коментари и препоръки – съвети за евентуални решения на откритите проблеми.

5. Начини за специфициране на изискванията. Видове модели в зависимост от перспективата на системата.

Моделът на процес е опростено описание на процес, обикновено представен от конкретна перспектива. Моделът отразява или възпроизвежда процеса на изследването в достатъчна степен, за да позволи получаване на информация за неговото изучаване.

5.1. Модел на потока на данните

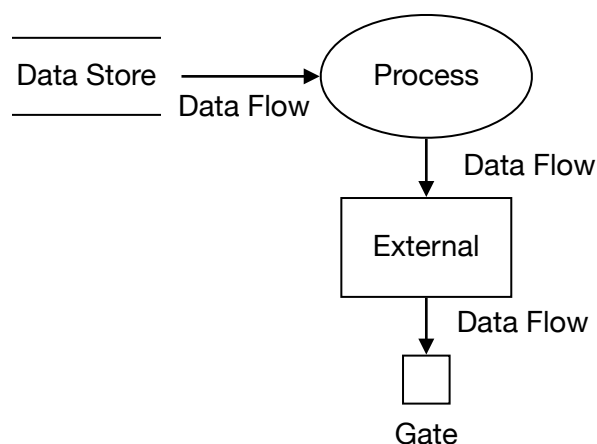
Също познат като Data Flow Model (DFM), той представя потока на информация в рамките на системата, както и между системата и средата. Той не показва последователност на поведение и на управляваща информация.

Диаграмата на моделът (DFD) е графично представяне на:

- Външни за процеса елементи;
- Процеси (функции);
- Поток на данните;
- Хранилища за данни.

Елементи на DFD:

- External Entity – външен източник или получател на данните. Те обикновено са извън обхвата на изучаваната област.
- Process – трансформира или манипулира данни;
- Data Flow – пренася данни между източник и процес, процес и получател или между два процеса;
- Data Store – хранилище на данни;
- Connector symbols – ползват се в случай, че диаграмата се представя на няколко страници.

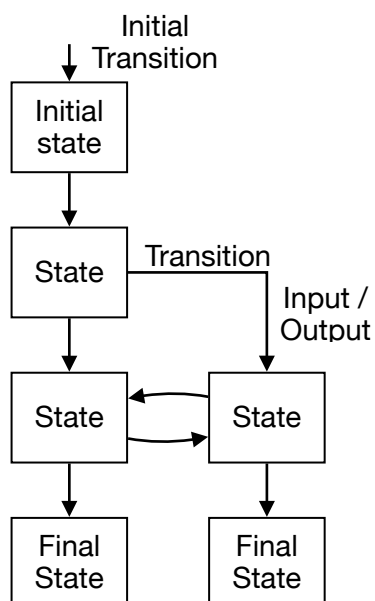


5.2. Модел на поведението

State Transition Diagram (STD) – показва динамиката в поведението на една система: Как системата реагира на последователност от външни събития и как независими компоненти координират работата си.

Основни компоненти:

- Състояние
- Действие/събитие – биват два вида:
 - външно и/или вътрешно
 - времево
- Преходи между състоянията

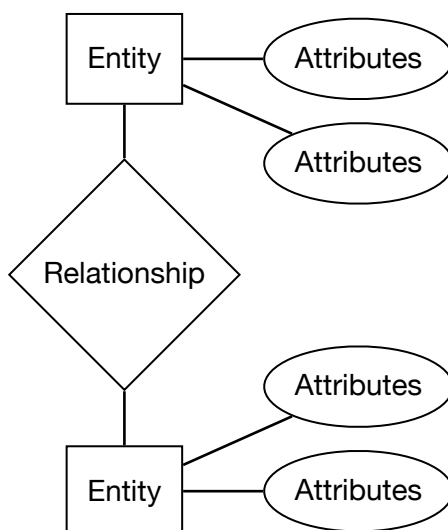


5.3. Семантични модели

Семантичният модел е модел на данните, който представя субектите в база данни, техните атрибути и техните връзки. Описва се чрез Entity Relationship Diagram-и (ERD), които представят логическия вид на данните и връзките между тях.

Основни компоненти:

- Същност (Entity): обект или дейност
- Атрибути (Attributes)
- Връзка (Relationships) – елементи:
 - кардиналност (cardinality): (1:1), (n:1), (1:n), (n:m)
 - опционалност: задължителна или незадължителна
 - зависимост



5.4. Обектно-ориентирани модели

ООМ представят едновременно данните и тяхната обработка, както и класификация и връзки на субектите на системата. Обектно-ориентираният анализ, проектиране и разработка е общоприет.

Основни концепции на ООМ:

- Обект – нещо, за което запазваме данни и описваме операции за манипулиране на тези данни.
- Клас – описание на множество от обекти, които споделят общи атрибути и поведение. Обектът е инстанция на клас.
- Операции – действия за четене и манипулиране на данните на обект.
- Метод – реализация на дадена операция от поведението на обекта. Специфицира се пътя, по който операциите се имплементират.
- Капсулиране на данни за обект – пакетиране на данни и операции, което предпазва данните на обекта от неразрешен достъп.
- Онаследяване – при таксономия на класовете обекти на по-ниско ниво в йерархията наследяват атрибутите и поведението на техните родители.
- Съобщения – начин по които обектите си комуникират. Когато обект получи съобщение, той изпълнява съответен метод.

5.5. Формални модели

Формалните модели осигуряват високо ниво на точност на системата, която да покрие напълно спецификацията, но не могат да гарантират абсолютна точност. Те са ниво на абстракция, която позволява да се създаде качествен софтуер.

Основни компоненти:

- Синтаксис – специфична нотация за представяне на спецификацията;
- Семантика – дефинира съвкупността от обекти, които ще се използват, за да се опише системата;

- Връзки – дефинират правилата, които показват кои обекти правилно удовлетворяват спецификацията.

