

Технически Университет — Габрово

Факултет: „Електротехника и електроника“

Катедра: „Компютърни системи и технологии“

Специалност „Компютърни системи и технологии“

РЕФЕРАТ

Тема: „Принципи и методи за проектиране на компютърни системи“

Изготвил: инж. Светлозар Владков Косев
ОКС: Магистър, Курс II
Фак. № 21605006

Проверил:
/ доц. В. Кукенска /

Габрово, 2017

неделя, 24 декември 2017 г.

Съдържание

Архитектура на компютърните системи. Основни модули	4
<i>Архитектура файл-сървър.....</i>	<i>4</i>
<i>Архитектура клиент-сървър</i>	<i>5</i>
Основни принципи.....	5
Модели	6
<i>Първо поколение (двуслоен модел)</i>	<i>7</i>
<i>Второ поколение.....</i>	<i>8</i>
<i>Интранет архитектура.....</i>	<i>9</i>
Системен подход при проектиране на компютърни системи и техните компоненти	11
<i>Системен подход</i>	<i>11</i>
<i>Характерни особености.....</i>	<i>12</i>
Системния подход като средство за организация.....	12
Творчески характер на използване на системния подход.....	13
Теоретични основи на системния подход	13
Експериментален характер	14
Прагматичност на системния подход	14
<i>Същност на системния подход.....</i>	<i>14</i>
Подходи за проектиране на системи и приложен софтуер.....	15
<i>Различни подходи за разработка на софтуер</i>	<i>15</i>
Основни подходи	15
<i>Структуриран.....</i>	<i>15</i>
<i>Гъвкав.....</i>	<i>15</i>
<i>Прототипен.....</i>	<i>16</i>
<i>Бързо разработване.....</i>	<i>16</i>
<i>Краен потребител.....</i>	<i>16</i>
<i>Комбиниран</i>	<i>16</i>
<i>Компютърно подпомогнато софтуерно инженерство</i>	<i>17</i>

Компютърно проектиран дизайн.....	17
Методи за инсталиране.....	18
Директен преход	18
Паралелен преход.....	18
Поетапен	18
Пилот.....	19
Съвместно проектиране на хардуера и софтуера на компютърна система.....	21
Използвана литература.....	25

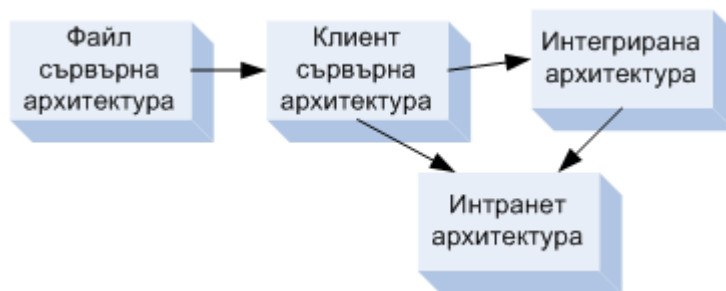
Фигури

Фигура 1 Архитектура на компютърни системи	4
Фигура 2 Общо хранилище на файлове.....	4
Фигура 3 Архитектура файл-сървър	5
Фигура 4 Основни процеси.....	6
Фигура 5 Модели	6
Фигура 6 Intranet система с използване средства на WWW	9
Фигура 7 Предимства на Intranet архитектура	10
Фигура 8 Системен подход.....	11
Фигура 9 Методи за внедряване на софтуер	19
Фигура 10 Риск срещу цената на методите за инсталиране на софтуер	20

Таблицы

Таблица 1 Трислоен модел.....	8
-------------------------------	---

Архитектура на компютърните системи. Основни модули

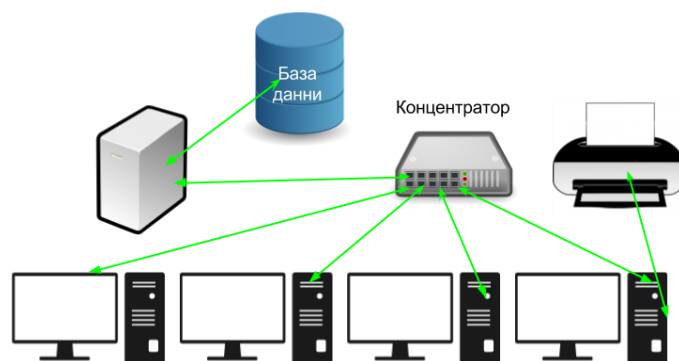


Фигура 1 Архитектура на компютърни системи

Архитектура файл-сървър

Използването на специализирани файл-сървъри все още се явява най-разпространената архитектура, използвана в малки информационни системи. Това се обуславя от:

- наличието на голямо количество компютри от различно ниво;
- сравнително евтино свързване на компютри в локални мрежи;
- възможността да се реализират информационни системи от не много опитни в областта на системното програмиране разработчици.



Фигура 2 Общо хранилище на файлове

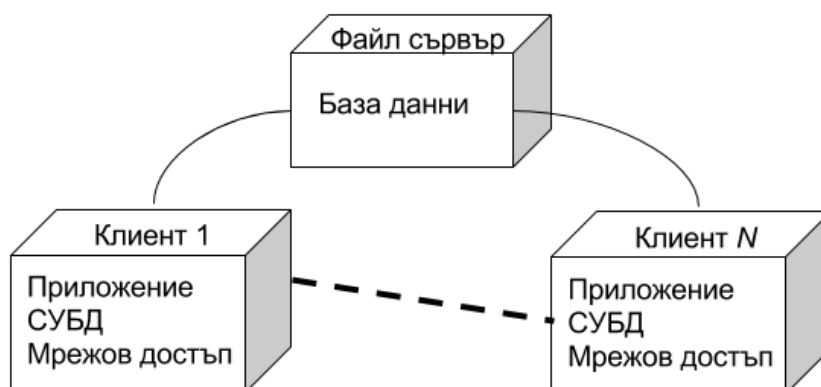
Съхранява се автономността на приложението и в голяма част на системното програмно осигуряване. Компонентите на система взаимодействат на база на общо хранилище на файлове, разположено на файл-сървъра. Във всеки

компютър се дублират не само приложните програми, но и средствата за управление на базите данни. Файл-сървърът представлява компютър с разширена дискова памет, използвана от всички абонати на мрежата.

Предимствата на тази архитектура са, че използва позната и удобна среда на Windows и наличие на прости за използване средства за разработка на база данни и съответни приложения.

Сървър за база данни е програмно осигуряване привързано към съответната база данни и съществуващо независимо от съответните потребителски приложения и изпълнявано на отделен компютър. Интерфейсът на сървърите за бази данни е базиран на езика за създаване на заявки към базите данни SQL като:

- от клиента към сървъра основно се изпращат оператори на езика SQL;
- от сървъра към клиента — резултатите от изпълнението на операторите.

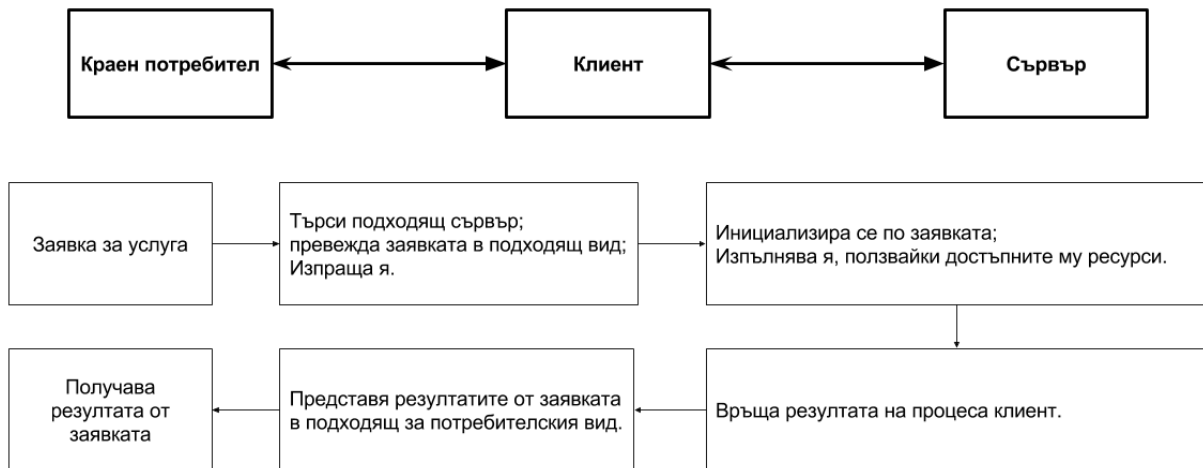


Фигура 3 Архитектура файл-сървър

Архитектура клиент-сървър

Основни принципи

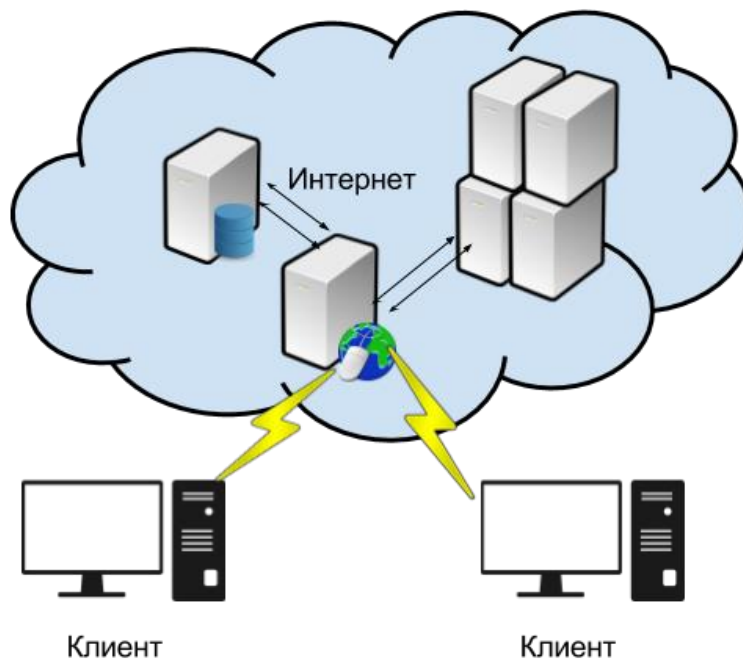
Услугите в мрежата се поддържат от два процеса в две точки на мрежата, които си комуникират чрез възлите между тях. Сървър — процесът във възела предоставящ услугата. Клиент — процесът във възела потребител на услугата.



Фигура 4 Основни процеси

Между двата процеса информацията протича през няколко слоя, свързани по строго специфични интерфейси, под управление на специфичен мрежов протокол.

Модели



Фигура 5 Модели

Първо поколение (двуслоен модел)

Двуслойният модел се базира на концепцията за клиент с богата функционалност. Целият код на приложението се намира върху работна станция – персонален или мрежов компютър. Върху сървъра се инсталира само системата за управление на базата данни (СУБД). Достъпът до базата данни се осъществява чрез езика SQL. Изпращат се заявки по мрежата от клиентите към сървъра. Отговорите на заявките също се връщат по мрежата.

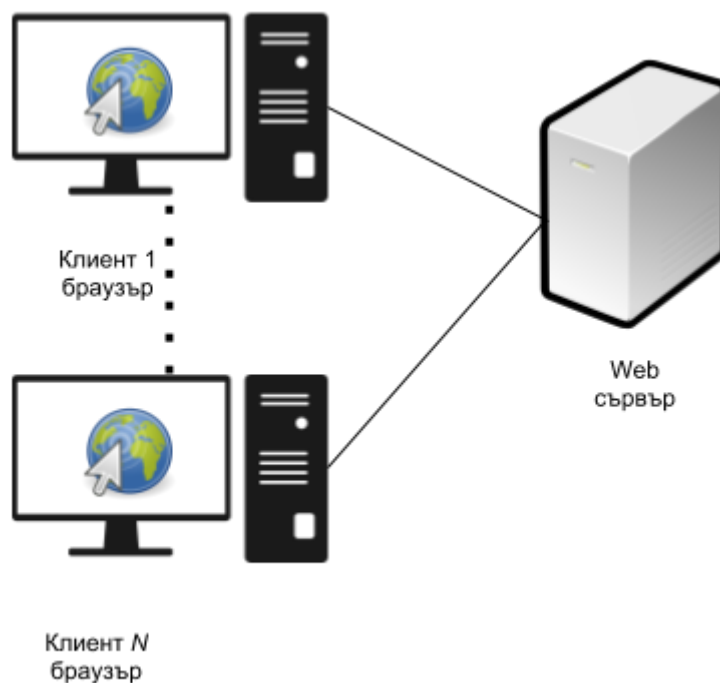
Второ поколение

Таблица 1 Трислоен модел

Проблем	Контрол на версиите при двуслойния модел
Причина	Всяка промяна в софтуера налага инсталирането на нов код върху стотици или дори хиляди компютри свързани към корпоративната информационна система. Системните администратори трябва да следят дали върху различните платформи е инсталирана една и съща версия на софтуерните модули.
Анализ	<ul style="list-style-type: none"> — Потребителския интерфейс и модела на данните рядко търпят изменения; — Бизнес правилата и бизнес логиката се променят постоянно.
Решение	Приложението се разделя на три компонента — интерфейс, приложна част и данни. Променливата бизнес логика се управлява от отделен сървър за приложения. Така модификациите в софтуера се извършват без да се налага инсталиране на нов код върху множеството от компютри.
Предимство	Въвеждането на приложно ниво като мост между графичния интерфейс на клиента и базата данни върху сървъра намалява стойността и сложността на поддръжката на софтуера.
Проблем	Производителността на системата намалява значително при нарастване броя на потребителите и обема на транзакциите.
Причина	Постоянния обмен на множество малки съобщения между графичния интерфейс на клиента и СУБД върху сървъра. Повечето локални мрежи са базирани на обща шина и конкурентен метод за достъп. При нарастване броя на потребителите нараства: <ul style="list-style-type: none"> — Броя на съобщенията, които трябва да се обменят; — Времето през което приложението чака в опашката за достъп до мрежовите.
Решение 1	Намаляване броя на съобщенията обменяни по мрежата. Постига се чрез обединяване в една машина сървърът за приложения и сървърът за бази данни.
Решение 2	Между сървърите и клиентите се включва комутатор, който разпределя честотната лента така, че да не се получават колизии.
Схема	<pre> graph LR subgraph Clients C1[Client 1] C2[Client 2] C3[Client 3] C4[Client 4] end subgraph Servers DS[Сървър за данни] AS[Сървър за приложения] end Concentrator[Концентратор] Switch[Комутатор] C1 --> Concentrator C2 --> Concentrator C3 --> Concentrator C4 --> Concentrator Concentrator <--> Switch Switch --> DS Switch --> AS </pre>

Интранет архитектура

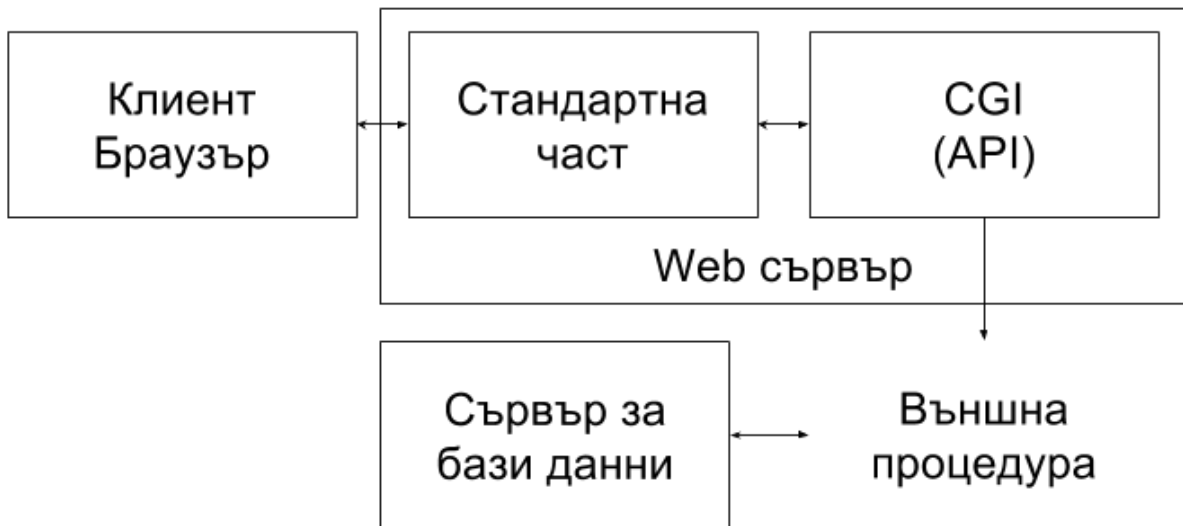
Развитието на Интернет услугите по естествен път оказва въздействие върху развитието на корпоративните информационни системи, чрез създаването на направление известно под названието Intranet. Видовете Интранет архитектури: локална (изолирана от Интернет) и виртуална корпоративна мрежа, интегрирана в Интернет. В този случай са особено важни средствата за защита на информацията от несанкциониран достъп.



Фигура 6 Intranet система с използване средства на WWW

Причините за използване на Intranet са, че езика HTML позволява сравнително лесно да се разработи удобна за използване структура, която ще се обслужва от един от стандартните Web сървъри. Също така възможността да се използва браузър избавя от необходимостта да се създава интерфейс с потребителя, като му предоставя удобни и развити механизми за достъп до информация. Предимствата на тази архитектура са проста организация, удобно използване, стандартен интерфейс и др. Ограниченията са, че системата не предоставя възможност за обработка на данни, а само разглеждане на

информация. Хипертекстовите структури трудно се модифицират. За да се промени съдържанието на Web сървъра е необходимо да са преустанови работата му и внесат необходимите промени. Не винаги търсенето на информация чрез хипервръзки, дава необходимите резултати. Понякога базата данни е наистина незаменима.

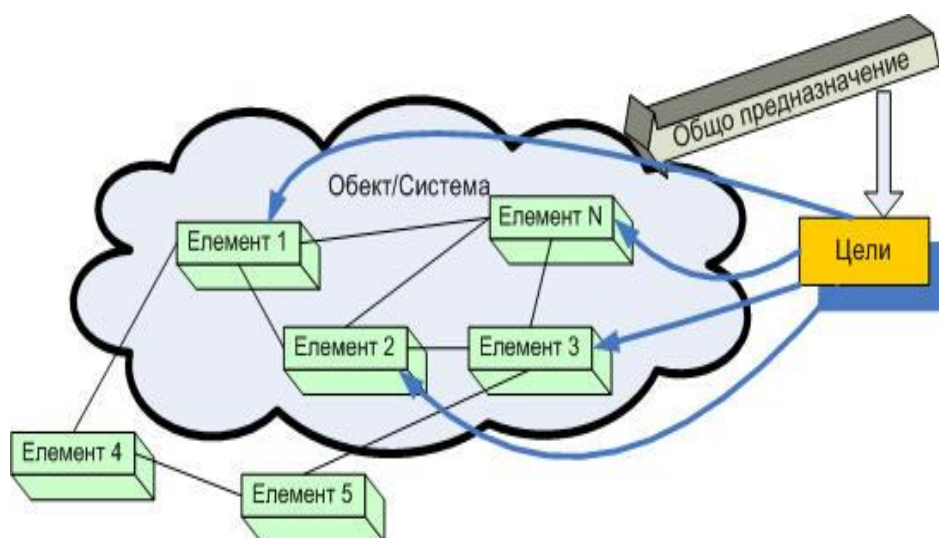


Фигура 7 Предимства на Intranet архитектура

Системен подход при проектиране на компютърни системи и техните компоненти

Системен подход

Системният подход е подход за изследване и управление на обекти, който ги разглежда като система, в която са определени елементите, вътрешните и външни връзки, влияещи на функционирането на системата, като целите на всеки елемент се формират в зависимост от общото предназначение на системата.



Фигура 8 Системен подход

Системният подход, който използваме за процесите на управление е разработен с цел прилагане на научен анализ в управлението на големите промишлени фирми. Той дава възможност да решим две основни задачи:

- свързани с разработване и управление на оперативни системи, т.е. системи, свързани с управление на различни дейности (система за управление на персонала; на обекти в реално време и т.н.);
- свързани с построяването на информационни системи, които се използват за вземане на управленски решения.

Между тези два типа задачи съществува взаимна връзка, която се състои в това, че построяването на информационна система дава възможност да се осигури помощ в процеса на взимане на решение от мениджърите на фирмата. Неговата основна концепция се заключава в това, че той разглежда управлението като един

процес по време на който се осъществява взаимодействие на частите (подсистемите на организацията). Този подход предвижда установяване на целите на системата и съсредоточаване на вниманието върху построяването на цялото за разлика от построяването на компонентите на системата. Главно значение при този подход има т.нар. синергетическа характеристика на системния подход (синергия — помощ; дружно действие на няколко части), т.е. организационните и информационните системи се строят по такъв начин, че да се постигне синергизъм, т.е. едновременно функциониране на отделни, но взаимно свързани части, осигуряващи по-висока ефективност в сравнение със сумарната ефективност на частите взети поотделно.

Характерни особености

Системния подход включва организационни, творчески, теоретически, експериментални и прагматични елементи.

Системния подход като средство за организация

Системният подход е средство за решаване на сложни и при това неясно формулирани проблеми, когато е необходимо организираното използване на голям брой ресурси. Обикновено група квалифицирани и способни на обобщения професионални експерти (създатели, проектантите на информационни системи) и специалисти (експерти по методи и компоненти) изучават всички обстоятелства в определен период от време с цел формулиране на проблемите. Последното има решаващо значение за целия проект на построяване на схемата, защото на базата на тази формулировка за проблемите могат да бъдат установени целите на бъдещата информационна система. Организационния подход изисква всички участници в проекта независимо от тяхната специална подготовка да са наясно със същността на системния подход.

Творчески характер на използване на системния подход.

Независимо от разработените общи процедури системния подход трябва да бъде един творчески процес, по време на който се отделя първостепенно внимание на целите, а чак след това на методите за постигане на тези цели. Окончателния вид на една бъдеща система ще зависи от творческите способности на отделните лица в екипа, от оригиналните решения, които те използват при нейното построяване. Причините, поради които системния подход трябва да носи творчески характер са следните:

- решаваните проблеми са изключително сложни, недостатъчно точно и ясно формулирани и по тази причина съществува една единствена формулировка (единствено решение);
- голяма част от изходните данни, с които разполагаме са непълни, неопределени и двусмислени, а това налага проектантът на системата да прояви известна доза въображение при решаване на проблема;
- за решаване на проблемите на отделните подчасти на фирмата трябва да бъдат намерени алтернативни решения и от тях да бъдат избрани тези, които осигуряват максимално приложение за системата като цяло;

традиционните функционални и отраслови бариери трябва да бъдат подчинени на интересите на синтез на решенията.

Теоретични основи на системния подход

В основата на системния подход лежат научни методи. Науката създава теоретичната основа върху която е базирано практическото решаване на проблемите. Различните данни, с които разполагат екипа проектанти може да осигури получаването на различни решения, но с помощта на теорията ще се определи методологиката, по която да бъде взето съответното решение.

Експериментален характер

Събирането на емпирични данни е съществена част от системния подход. В този процес трябва да отделим недостоверните данни от достоверните, свързаните с проблема от несвързаните с проблема. Обикновено събраните факти включват в себе си както факти, свързани с техническите аспекти на проблема, така и факти свързани с практиката и други характеристики на съответната организация.

Прагматичност на системния подход

При практическото построяване на реални информационни системи най-важната характерна особеност на системния подход е, че той осигурява получаването на резултати, които са приложими за реалната дейност на съответната система. Създадената информационна система трябва да бъде достъпна, производителна и управляема. Тя трябва да осигурява постигането на практически цели или да удовлетворява реални ценности. По тази причина проектантите на информационни системи трябва да са наясно с характера на дейността на фирмата, за която се изгражда съответната информационна система. Системния подход се състои от два основни етапа: етап на разширение или анализ и етап на свиване или синтез.

Същност на системния подход

Системният подход формулира целите и изяснява тяхната йерархия до началото на всяка дейност, свързана с управлението, особено с вземането на решения. Реализира поставените цели при минимални разходи, чрез използване на сравнителен анализ на алтернативните пътища и методи и чрез осъществяване на съответен избор. Прави количествена оценка на целите, методите и средствата за тяхното достигане, базирана на всеотранно изучаване на възможните и планираните резултати от дейността.

Подходи за проектиране на системи и приложен софтуер

Различни подходи за разработка на софтуер

Всички подходи за разработка на софтуер използват пет стъпки на цикъл на разработка на софтуер. Тези стъпки са както следва:

- определяне и разбиране на проблема;
- планиране и проектиране;
- изпълнение;
- тестване и оценка;
- поддръжка.

Основни подходи

Структуриран

Този подход, както подсказва името, включва много структурирани етапи стъпка по стъпка. Всеки етап от цикъла на разработка на софтуер трябва да бъде завършен, преди да продължите към следващата стъпка. Структурираният подход признава, че етапите, свързани с разработката на софтуер, трябва да се осъществяват систематично. С други думи, не може да се приложи решение, докато не се разгледат потенциалните решения, и потенциалните решения не могат да се предвидят, докато не се получи проблема напълно дефиниран и разбран.

Гъвкав

Възникна в отговор на реалността на много проекти за разработка на специален софтуер. Акцентира се върху екипа, развиващ системата, вместо да следва предварително дефинирани структурирани процеси. Премахва необходимостта от подробни изисквания и сложна документация за проектиране.

Прототипен

Създава прототипи, като подход за разработка на софтуер. Поддава се на интензивно взаимодействие между клиенти и разработчици на софтуер. Това е кръгов подход към разработката на софтуер, където през целия процес на разработка клиентът успешно валидира продукта. Често прототипът постепенно се превръща в крайния продукт. Всеки последователен прототип отговаря по-добре на оригиналните и реализирани изисквания за крайния продукт.

Бързо разработване

Подходът за бързо разработване на приложения RAD (**R**apid-**A**pplication **D**evelopment), както подсказва името, е подход за разработка на софтуер, който позволява използването на използваеми системи в рамките на малко време. Подходът е въведен, за да преодолеят дългите закъснения, които се срещат при клиентите, когато се използват по-конвенционални методи.

Краен потребител

Подходът на крайния потребител на разработката на софтуер става все по-популярен, тъй като пакетите на приложения са пуснати на пазара, за да бъдат персонализирани от потребители с ограничен опит в разработването на софтуер. Този подход на крайния потребител включва фирми, които създават свои собствени продукти, като персонализират други приложения, използвайки помощни програми и други автоматични устройства за генериране на кодове

Комбиниран

В някои аспекти на решението могат да се придадат на повече от един подход на развитие отколкото на друг. Крайната цел на разработката е да се създаде подходящо документирано решение на проблема.

Компютърно подпомогнато софтуерно инженерство

Компютърно подпомогнатото софтуерно инженерство CASE (Computer-Aided Software Engineering) е домейнът на софтуерните инструменти, използвани за проектиране и внедряване на приложения. Инструментите му са сходни и частично вдъхновени от инструментите на компютърно проектирания дизайн (CAD), използвани за проектиране на хардуерни продукти. CASE инструменти се използват за разработване на висококачествен, безпроблемен и поддържан софтуер. Софтуерът често се свързва с методи за разработване на информационни системи, заедно с автоматизирани инструменти, които могат да бъдат използвани в процеса на разработване на софтуер.

Много интегрирани инструменти на компютърно подпомогнатото софтуерно инженерство осигуряват всички или някои от следните функции:

- системно моделиране;
- създаване на речник на данни;
- изготвяне на документация;
- създаване на автоматични тестови данни;
- софтуерно версиране;
- производство на шаблони за изходния код.

Например Visual Studio има инструменти, някои от които включват шаблони за различни типове проекти, мениджър на кодови фрагменти, система за версии и различни инструменти за отстраняване на грешки.

Компютърно проектиран дизайн

Компютърният дизайн (CAD) е използването на компютърни системи или работни станции за подпомагане на създаване, модифициране, анализ и оптимизиране на дизайн. CAD софтуерът се използва за повишаване на производителността на дизайнера, подобряване на качеството на дизайна, подобряване на комуникациите чрез документация и създаване на база данни за производство. Изходът на CAD често е под формата на електронни файлове за

печат, обработка или други производствени операции. Използва се и терминът CADD (за компютърно проектиране и проектиране).

Методи за инсталиране

Съществуват четирите общи метода за инсталиране на нови или актуализирани системи:

Директен преход

Този метод включва старата система, която напълно се пропуска, а новата система се изпълнява едновременно. Старата система става вече невалидна.

Паралелен преход

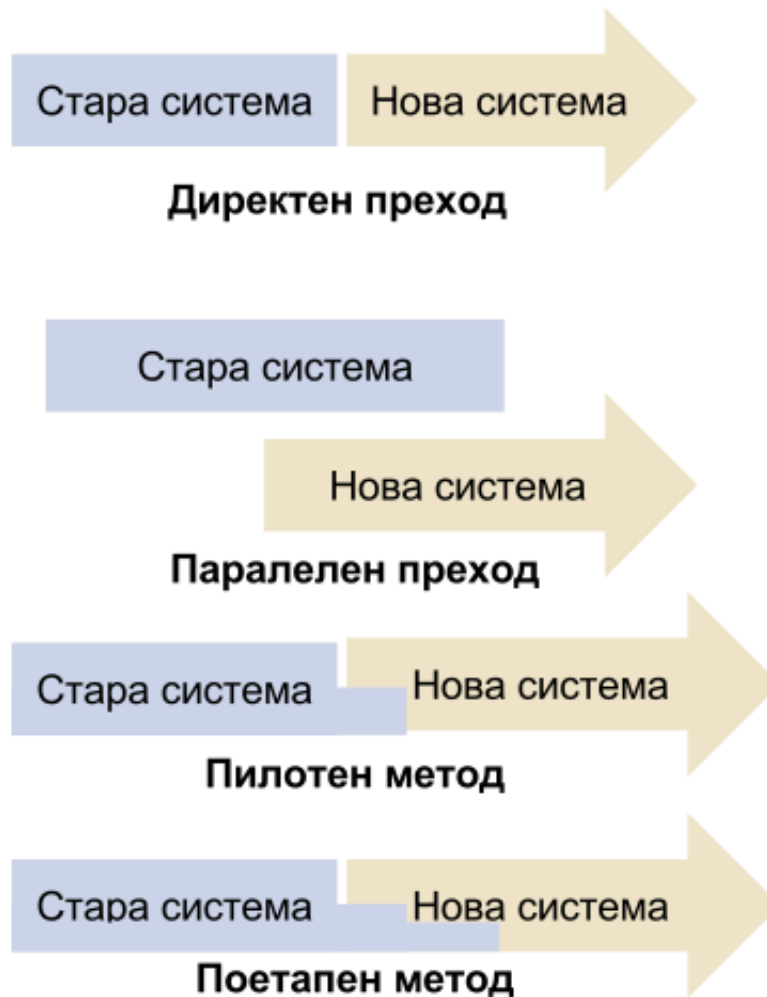
Паралелният метод на изпълнение (или преобразуване) включва работата на двете системи заедно за определен период от време. Това позволява да се срещнат и коригират сериозни проблеми с новата система без загуба на данни. Паралелното преобразуване означава също така, че потребителите на системата пестят време, за да се запознаят напълно с работата на новата система. По същество старата система остава оперативна като резервна за новата система. След като се установи, че новата система отговаря на изискванията, тогава работата на старата система може да се прекрати. Паралелният метод включва двойно натоварване за потребителите, тъй като всички функции трябва да се изпълняват както на старата, така и на новата система.

Поетапен

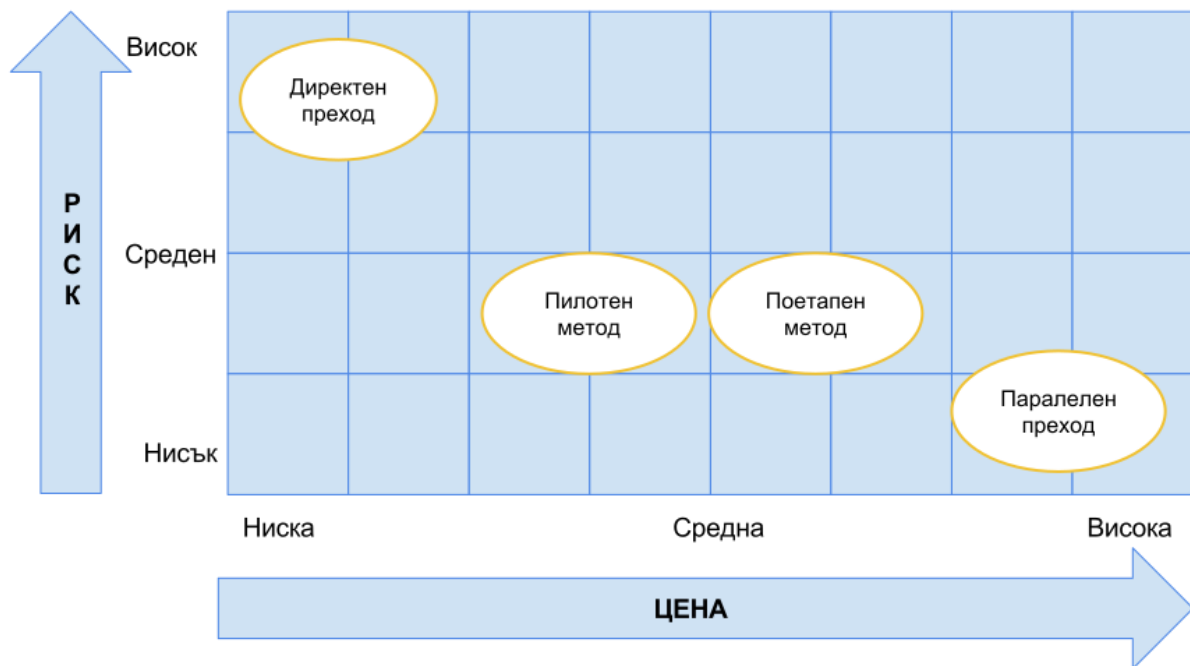
Поетапният метод за внедряване (или преобразуване) на стара система в нова система включва постепенно въвеждане на новата система, докато старата система постепенно се изоставя.

Пилот

С пилотния метод на внедряване (или преобразуване), новата система е инсталирана за малък брой потребители. Тези потребители научават, използват и оценяват новата система. След като се приеме, че новата система изпълнява задоволително, системата се инсталира и използва от всички.



Фигура 9 Методи за внедряване на софтуер



Фигура 10 Риск срещу цената на методите за инсталиране на софтуер

Съвместно проектиране на хардуера и софтуера на компютърна система

Най-ранните вградени системи са системи за обработка на банкови и транзакции, работещи на мейнфрейм и масиви от дискове. Дизайнът на такава система предполага съвместно проектиране (кодизайн) на хардуер и софтуер: предвид очаквания брой и вид транзакции, които трябва да се извършат в рамките на един ден, трябва да бъде избрана хардуерна конфигурация, която да поддържа очаквания трафик и да се създаде софтуерен дизайн за ефективно.

Създаването на вградена компютърна система, която отговаря на нейните цели за изпълнение, разходи и проектиране, е проблем на съвместно проектиране на хардуер и софтуер — дизайнът на хардуерните и софтуерните компоненти се влияе един друг. Докато вградените системи варират от микровълнови фурни до системи за управление на въздухоплавателни средства, има общи техники за проектиране на тези различни приложения. Освен това дизайнът на вградената система често изисква техники, които са малко по-различни от тези, които се използват или за проектирането на универсални компютри, или за приложния софтуер, работещ на тези машини. Вграденото изчисление е уникално, защото е проблем при съвместното проектиране на хардуер и софтуер — хардуерът и софтуерът трябва да бъдат проектирани заедно, за да се гарантира, че внедряването не само функционира правилно, но също така отговаря на целите за ефективност, разходи и надеждност.

Дизайнът на вградените системи не е толкова добре разбран, колкото и проектирането на интегрални схеми, които имат няколко методологии за различни разходно-ефективни сделки — морски порти, стандартни клетки, инструменти с пълна готовност и дизайн за етапите на проектиране в всяка методология. Въпреки че дизайнерите на вградените системи могат да използват съществуващите инструменти за хардуерни и софтуерни компоненти, след като дизайнът е бил разделен, остават още много неща за това как една система е разделена на хардуерни и софтуерни компоненти. Методологиите и инструментите за съвместно проектиране на хардуер и софтуер са критични изследователски теми за дизайн на вградената система.

Съвместното проектиране на вградените хардуерно-софтуерни системи трябва да се извършва на няколко различни нива на абстракция, но най-високите нива на абстракция в кодизайна са по-абстрактни от типичния софтуер програмист или дизайнер на специфична за приложението интегрална схема (ASIC) може да се използва. Критичните архитектурни решения се вземат с помощта на абстрактни хардуерни и софтуерни елементи: процесори, памет и процеси. В резултат на това първоначалните проблеми с хардуера и софтуера са на високо ниво: първото решение за хардуерно проектиране е да се изгради мрежа от процесори, памет и периферни устройства; първият проблем с дизайна на софтуера е да се разделят необходимите функции в комуникационните процеси. Най-напред, хардуерния дизайнер може да не сметне, че изборът на процесор е много важно. Например, основното архитектурно решение на хардуера може да бъде избор между 32- или 64-битов компютър. Тази задача обаче не е толкова различна от дизайнерските решения, пред които са изправени дизайнерите на многомащабна интеграция (VLSI).

Дизайнът на софтуерната архитектура — разделянето на функциите в комуникационните процеси — е тясно свързан с дизайна на двигателя. Две системи със същите функции, но с различни структури на процесите, могат да работят при много различни скорости, да изискват много различно количество памет и т.н. Например в един случай разделянето на един процес на два може да увеличи количеството на съпротивлението, изложено в системата и да намали изисквано време за изпълнение. В друг случай, разделянето на задачата на много процеси може да доведе до прекалено много контекстни превключватели и да намали ефективността. Не можете да се избере хардуерен двигател, без да се избере софтуерна архитектура, тъй като структурата на процеса на софтуера помага да определите размера и скоростта. По подобен начин тялото и аеродинамиката на превозното средство са тясно свързани с избора на двигателя: тесният самолет не може да побере широк двигател. Дизайнът на вградената компютърна система е съвместно проектиране на хардуер-софтуер точно поради това, че дизайнът на системната архитектура трябва едновременно да отчита хардуерния механизъм и структурата на софтуера.

Процесът на проектиране на вградена система трябва да се различава значително от приложението: дизайнът на телефон е много различен от този на автопилота. Може обаче да бъдат определени общи стъпки. Освен това, изследване на типичен процес на проектиране показва, че хардуерните и софтуерните компоненти на вградената система имат общи абстракции, факт, който може да се използва в наша полза при съвместно проектиране на хардуер и софтуер.

Фактът, че хардуерът и софтуерът са определени като процеси, ни дава надежда, че може да се използва обща техника за моделиране, за да проектиране едновременно хардуерния двигател и кода на приложението, за да отговорим на ограниченията на производителността. Това е така, защото хардуер и софтуер имат свързани абстракции, че съвместното проектиране може да направи архитектурни решения, които балансират дизайнерските проблеми на всеки от тях.

След като са направени архитектурни решения, дизайнът на хардуера и софтуера може да продължи до известна степен поотделно — ако правилното съвместно проектиране е избрало добра архитектура, тогава компонентите, проектирани според архитектурните спецификации, могат да бъдат събрани за изграждане на задоволителна система. Проектирането на хардуера се извършва чрез няколко стъпки:

- описание на поведението, което може да включва комуникационни машини и в които операциите са частично насрочени във времето;
- конструкция на трансфер на регистър, която дава комбинирани логически функции между регистрите, но не и детайлите на логическата конструкция;
- самият логически дизайн;
- физическия дизайн на интегрална схема, разположението и маршрутизирането в полевопрограмируемо логическо устройство и т.н.

Софтуерният дизайн започва с набор от комуникационни процеси, тъй като повечето вградени системи имат времево поведение, което най-добре се изразява като едновременна система. Разграждането на функцията в модули е междинна стъпка в дизайна на софтуера, която води до кодиране в комбинация от монтажни

и езикови езици на високо ниво. Софтуерните и хардуерните компоненти трябва да бъдат интегрирани и след това тествани, за да се увери, че системата отговаря на техните спецификации.

Симулацията ще бъде важен инструмент за съвместно проектиране в обозримо бъдеще, поради сложния характер на вградените изчислителни системи. Тъй като компонентите на вградените системи са толкова сложни, може да е трудно да се разработят изчерпателни аналитични модели за тяхното изпълнение. Симулацията също така помага на проектанта да провери дали системата отговаря на неговите изисквания. Съ-симулацията смесва компоненти, които имат различни симулационни модели. Съвместната симулация обикновено се отнася до някакъв вид смесена хардуерно-софтуерна симулация — например, една част от системата може да бъде моделирана като инструкции, изпълняващи се от процесора, докато друга част може да бъде моделирана като логически портове. Ко-симулацията е трудна, защото компонентите на системата работят на различни нива на абстракция — аналоговите компоненти работят над напрежение, PLD (програмируеми логически устройства) работят над двоични стойности и микропроцесорите работят по инструкции — и работят с различни скорости, една инструкция в микропроцесора може да отнеме няколко цикъла изпълни, през което време аналоговите компоненти може да са се преместили в драстично състояние.

Използвана литература

1. Системен подход — <http://tuj.asenevtsi.com/Informatica2/I051.htm> — дата на ползване: декември, 2017;
2. System design — https://en.wikipedia.org/wiki/Systems_design — дата на ползване: декември, 2017;
3. Definition of Systems Design — <https://economictimes.indiatimes.com/definition/systems-design> — дата на ползване: декември, 2017;
4. Application of software development approaches — <https://www.purplezeus.com/application-of-software-development-approaches.html> — дата на ползване: декември, 2017;
5. Computer-aided software engineering — https://en.wikipedia.org/wiki/Computer-aided_software_engineering — дата на ползване: декември, 2017;
6. Computer-aided design — https://en.wikipedia.org/wiki/Computer-aided_design — дата на ползване: декември, 2017;
7. Проектиране на Компютърни системи — <http://www.referati.org/proektirane-na-ks/43089/ref> — дата на ползване: декември, 2017;