**Процес на разработка на софтуер**

**Основи на процеса на разработка на софтуер**

Терминът "софтуерно инженерство" се появява за първи път през 1968 г. на конференция на НАТО и има за цел да стимулира търсенето на решения на "софтуерната криза", която се случва по това време. Оттогава се превърна в професия на софтуерен инженер и изследователска област, посветена на това да направи софтуера по-добър, по-достъпен, по-добре поддържан и по-бърз за разработка. Разработката на софтуер е свързана с дисциплините компютърни науки, управление на проекти и системно инженерство.

**Етапи в разработката на софтуер**

Планирането е част от всеки проект. В процеса на планиране се откриват конкретни задачи свързани със самия проект.

След като основните изисквания са събрани от клиента, започва техния по-задълбочен анализ. Определя се обхвата на разработения продукт като се поставят конкретни задачи на проекта и се изработва съответната документация.

Някои функционалности могат да останат извън обхвата на проекта като впоследствие могат да го оскъпят. Най-честа причина е неяснота по отношение на изискванията и приложението в самото начало на разработване. Ако друга компания извършва разработването и планирането, този документ може да се счита за правен документ и е част от договорните отношения. В случай на възникване на спорове и двусмислено тълкуване – какво е обещано на клиента и какво е получено като продукт, документацията по разработване на проекта може да се приложи за изясняване и разрешаване на спорни моменти.

**Методологии за разработка на софтуер**

Софтуерна методология при писането на [софтуер](https://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80) е рамка, която се използва за структуриране, планиране и контролиране процеса на разработка на [информационни системи](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B8_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8). През годините са еволюирали много разновидности на такива платформи, всяка със своите отличителни предимства и недостатъци. Има няколко различни подхода в разработката на софтуер: някои използват по-структуриран, инженерен подход за разработка на бизнес решения, докато други възприемат по-частични подходи, като софтуерът се развива на части. Една системна методология на разработка на софтуер не винаги е подходяща за всички проекти. Всяка от възможните методологии е най-подходяща за определен тип проекти, в зависимост от различните технически, организационни, проектни и екипни спецификации.

Повечето методологии споделят някои от следните комбинации в разработката на софтуер:

* Анализиране на потребностите
* Проучване на пазара
* Събиране на изискванията за предложеното бизнес решение
* Изработване на план или дизайн на софтуерно решение
* Анализиране на проблема (математически проблем, задача)
* Писане на програмен код (имплементация)
* Тестване на софтуера и оправяне на [бъгове](https://bg.wikipedia.org/wiki/%D0%91%D1%8A%D0%B3)
* Внедряване
* Поддръжка и оправяне на нови [бъгове](https://bg.wikipedia.org/wiki/%D0%91%D1%8A%D0%B3)

Тези етапи често се свързват със софтуерната разработка. Различните подходи в разработката на софтуер ги използват в различна последователност или посвещават повече или по-малко време на някои от тях. Детайлите в документацията на всеки етап също може да варират. Посочените етапи могат да се свържат с подхода на „водопада“ или може да бъдат повтаряни в различни цикли. Подхода с различни цикли обикновено включва по-малко отделено време за планиране и документиране, но повече време в писане на програмен код и разработка на автоматизирани тестове. Различните цикли предлагат също непрекъснато тестване по време на живота на разработка, както и работещ (или без бъгове) продукт през цялото време. По- структурираните подходи на „водопада“ се стремят да оценяват рисковете и да разработят детайлен план преди писането на програмен код, за да се избегнат значителни промени в дизайна и програмния код на по-късен етап.

Кой е най-добрият подход често зависи от типа на проблема. Ако проблемът е добре разбран и може да бъде планирано решение за дълъг период от време напред, то подходите на „водопада“ често са най-удачни. От друга страна, ако проблемът е уникален (или е такъв поне за екипа от разработчици) и структурата на софтуерното решение не може да бъде лесно визуализирана, то тогава „по-екстремните“, частични подходи биха свършили по-добра работа.

**Софтуерни изисквания**

Софтуер за управление на продажбите в търговски обект (СУПТО), включен в [списъка по по чл. 118](https://staging.egov.bg/wps/portal/nra/nra/fiskalni-ustroystva-supto-i-e-magazini/supto/page.Iziskvania-kum-softuera/!ut/p/z1/jZDJDoJAEES_xQNH6TYzKnojiuCWjAuKfTFgcCBBxiBK_HuJnhS3vlXnVS8FBB5Q6l9i6eexSv2k1BtqbYWwTcc2EDts0ccZW_SafO6w1ZTD-gWYTBjOxJj1x0sLLbsJ9I8fP5SJ__m_APR9_BroZcWK8_KDjiGsaSlFowII27oDfDgYNJC1K0A1g19XjIBkooJH4GYaMEMCZeE-zMJMP2dlO8rz46mroYZFUehSKZmE-k4dNHxnidQpB--ZhOPBdT2Mh3UKroVZq90AsiphXA!!/?1dmy&urile=wcm%3apath%3a%2Fagency%2Fsite%2Fregisters-i-spisuci%2Fspisuci%2Fpublichen-elektronen-spisuk-na-supto%2Fpublichen-elektronen-spisuk-na-supto), ал. 16 от ЗДДС, трябва да отговаря на изискванията съгласно [приложение №29](https://staging.egov.bg/wps/portal/nra/nra/fiskalni-ustroystva-supto-i-e-magazini/supto/page.Iziskvania-kum-softuera/!ut/p/z1/jZDJDoJAEES_xQNH6TYzKnojiuCWjAuKfTFgcCBBxiBK_HuJnhS3vlXnVS8FBB5Q6l9i6eexSv2k1BtqbYWwTcc2EDts0ccZW_SafO6w1ZTD-gWYTBjOxJj1x0sLLbsJ9I8fP5SJ__m_APR9_BroZcWK8_KDjiGsaSlFowII27oDfDgYNJC1K0A1g19XjIBkooJH4GYaMEMCZeE-zMJMP2dlO8rz46mroYZFUehSKZmE-k4dNHxnidQpB--ZhOPBdT2Mh3UKroVZq90AsiphXA!!/?1dmy&urile=wcm%3apath%3a%2Fagency%2Fsite%2Fzakonodatelstvo%2Fdanutsi%2Fnaredbi%2F176b939c-0d90-49d9-9102-ea2cf4495c17) на [Наредба №Н-18/2006 г.](https://staging.egov.bg/wps/portal/nra/nra/fiskalni-ustroystva-supto-i-e-magazini/supto/page.Iziskvania-kum-softuera/!ut/p/z1/jZDJDoJAEES_xQNH6TYzKnojiuCWjAuKfTFgcCBBxiBK_HuJnhS3vlXnVS8FBB5Q6l9i6eexSv2k1BtqbYWwTcc2EDts0ccZW_SafO6w1ZTD-gWYTBjOxJj1x0sLLbsJ9I8fP5SJ__m_APR9_BroZcWK8_KDjiGsaSlFowII27oDfDgYNJC1K0A1g19XjIBkooJH4GYaMEMCZeE-zMJMP2dlO8rz46mroYZFUehSKZmE-k4dNHxnidQpB--ZhOPBdT2Mh3UKroVZq90AsiphXA!!/?1dmy&urile=wcm%3apath%3a%2Fagency%2Fsite%2Fzakonodatelstvo%2Fzakonodatelstvo_priority%2F238edf96-5722-4993-9eeb-1ab7082e6e7b) - чл. 52а, ал. 1.

Допуска се софтуер за управление на продажби, включен в списъка по чл. 118, ал. 16 от ЗДДС, да отговаря най-малко на изискванията по т. 1, 2, 4, 6 и 7 от приложение № 29, при условие че се управляват продажби само на стоки или услуги, заплащани напълно или частично от Министерството на здравеопазването или от Националната здравноосигурителна каса, съгласно националните рамкови договори за медицински дейности и за дентални дейности или индивидуалния договор за заплащане на лекарствени продукти по чл. 262, ал. 6, т. 1 от Закона за лекарствените продукти в хуманната медицина, на медицински изделия и на диетични храни за специални медицински цели, сключен по реда на Закона за здравното осигуряване, или заплащани чрез предоставените субсидии и финансирания по реда на § 1, т. 15 от допълнителните разпоредби на ЗДДС.

Допуска се чрез софтуера да се управляват продажби и на други медицински услуги, при условие че за тях се отразява само медицинска информация без посочена стойност. В този софтуер не се допуска да се управляват други продажби извън изрично посочените - чл. 52а, ал. 2.  
Допуска се софтуер за управление на продажбите,  включен в списъка по чл. 118, ал. 16 от ЗДДС, да отговаря само на изискванията по т. 2, 4, 6 и 7 от Приложение № 29, само когато се използва от лица, отговарящи на условията, посочени чл. 52а1 на Наредба №Н-18/2006 г.

**Случай на употреба (use cases)**

Случай на употреба е списък от действия или стъпки от събития, обикновено определящи взаимодействията между дадена роля и система за постигане на цел. Актьорът може да бъде човек или друга външна система. В системното инженерство случаите на използване се използват на по-високо ниво, отколкото вътре софтуерно инженерство, често представляващи мисии или заинтересована страна цели. След това подробните изисквания могат да бъдат уловени в Език за моделиране на системи (SysML) или като договорни изявления.

**Потребителски истории user stories**

Потребителска история е неформално естествено езиково описание на една или повече функции на софтуерна система. Потребителските истории често се пишат от гледна точка на краен потребител или потребител на система. Те често се записват на индексни карти, на Лепящи се листчета или цифрово в софтуера за управление на проекти[1] В зависимост от проекта потребителските истории могат да бъдат написани от различни заинтересовани страни, включително клиенти, потребители, мениджъри или членове на екипа за разработка.

**Спецификация на софтуерните**

Спецификация на софтуерните изисквания (SRS) е описание на a софтуерна система да бъде разработен. Моделира се след това спецификация на бизнес изискванията, известен също като a спецификация на изискванията на заинтересованите страни (StRS). Излага се спецификацията на софтуерните изисквания функционален и нефункционални изисквания и може да включва набор от случаи на употреба които описват потребителските взаимодействия, които софтуерът трябва да предоставя на потребителя за перфектно взаимодействие.

**Прототипиране**

Използваните методи за проектиране на интерфейс са основно два – чрез използване на [прототипи](https://bg.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF) или [симулация](https://bg.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D1%83%D0%BB%D0%B0%D1%86%D0%B8%D1%8F).

Проектирането на типичен човеко-машинен интерфейс се състои от следните етапи: спецификация на взаимодействието, спецификация на софтуера, създаване на прототип, като за всеки етап са присъщи следните основни практики:

при спецификацията на взаимодействието: фокусиран върху потребителя дизайн; ориентиран към дейността дизайн; дизайн, базиран на сценарий; гъвкав дизайн.

при софтуерната спецификация: използване на прецеденти, налагане на ограничения чрез протоколи за взаимодействие (с цел да се избегнат грешки).

при създаването на прототип: [интерактивен](https://bg.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%B5%D0%BD) дизайн, който от своя страна ползва библиотеки от интерфейс елементи (контроли, декорации и т.н.).

Източници:

<https://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80%D0%BD%D0%BE_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D1%81%D1%82%D0%B2%D0%BE>

<https://bg.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BD%D0%B0_%D1%81%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80>

<https://staging.egov.bg/wps/portal/nra/nra/fiskalni-ustroystva-supto-i-e-magazini/supto/page.Iziskvania-kum-softuera>

<https://ewikibg.top/wiki/use_case>

<https://wikibgbg.top/wiki/User_story>

<https://wikibgbg.top/wiki/Software_requirements_specification>

<https://bg.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D1%80%D0%B5%D0%B1%D0%B8%D1%82%D0%B5%D0%BB%D1%81%D0%BA%D0%B8_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81#%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%B0%D0%BD%D0%B5>

Александър Бинчев 12а