

Доставка на софтуер - особености и процеси

Тема №2

Подходи при доставката на софтуер

Част 1: Модели на внедряване и канали за доставка

Съдържание

1	Въведение	3
2	Модел на внедряване	3
3	Канал за доставка	4
4	On-premise софтуер	5
4.1	Предимства	5
4.1.1	Контрол и сигурност	5
4.1.2	Предвидимост на натоварването	5
4.1.3	Конфигуриране	5
4.2	Недостатъци	5
4.2.1	Разходи за оборудване и поддръжка	5
4.2.2	Скалируемост	6
4.3	Процес на доставка	6
4.3.1	Гледна точка на доставчика	6
4.3.1.1	Планиране и разработка	6
4.3.1.2	Избор на канал за доставка	6
4.3.1.3	Подготовка за физическата доставка	6
4.3.1.4	Физическата доставка	7
4.3.1.5	Поддръжка и актуализации	7
4.3.2	Гледна точка на консуматора	7
4.3.2.1	Планиране и изисквания	7
4.3.2.2	Придобиване на софтуера	7
4.3.2.3	Подготовка на инфраструктурата	7
4.3.2.4	Внедряване	8
4.3.2.5	Тестване и оптимизация	8
4.3.2.6	Поддръжка и актуализации	8
4.4	Сигурност	8
4.5	Скалируемост и оптимизация	9
4.5.1	Скалиране	9
4.5.2	Оптимизация на ресурсите	9
4.6	Актуализации	9
5	Cloud софтуер	10
5.1	Предимства	10
5.1.1	Скорост и гъвкавост в развитието	11
5.1.2	Намалени разходи за оборудване и поддръжка	11
5.1.3	Скалируемост и гъвкавост в ресурсите	11
5.1.4	Глобална достъпност	11
5.1.5	Автоматизация и управление на ресурсите	11
5.1.6	Бързо въвеждане на нови функции и актуализации	11
5.2	Недостатъци	11

5.2.1	Зависимост от Интернет свързаността	12
5.2.2	Въпроси за сигурност и контрол	12
5.2.3	Сложност на интеграция със съществуващи системи	12
5.2.4	Ограничения на контрола и настройките	12
5.2.5	Преносимост на данни	12
5.3	Процес на доставка	12
5.3.1	Планиране	12
5.3.2	Избор на облачен доставчик и конфигурация	12
5.3.3	Разработка	13
5.3.4	Тестване и оптимизация	13
5.3.5	Избор на канал за доставка	13
5.3.6	Подготовка за физическа доставка	13
5.3.7	Физическа доставка	13
5.3.8	Поддръжка и актуализация	14
5.3.9	Мониторинг и управление на ресурсите	14
5.4	Сигурност	14
5.4.1	Защита на данни и достъп	14
5.4.2	Реакция при инциденти и бекъп решения	14
5.4.3	Обучение и осведоменост	14
5.5	Скалируемост и оптимизация	14
5.5.1	Скалиране на Cloud решения	15
5.5.2	Оптимизация на ресурсите в облака	15
5.5.3	Балансиране на натоварването	15
5.6	Актуализации	15
5.6.1	Защо са необходими редовни актуализации?	15
5.6.2	Управление на актуализациите	16
5.6.3	Актуализации от облачния доставчик	16
6	Mobile софтуер	16
6.1	Предимства	16
6.2	Недостатъци	17
6.3	Процес на доставка	17
6.3.1	Планиране	18
6.3.2	Разработка	18
6.3.3	Тестване и оптимизация	18
6.3.4	Избор на канал за доставка	18
6.3.5	Подготовка за физическа доставка	18
6.3.6	Физическа доставка	18
6.3.7	Поддръжка и актуализация	18
6.4	Сигурност	18
6.5	Скалируемост и оптимизация	19
6.6	Актуализации	19
7	Заключение	20

1 Въведение

В тази лекция ще разгледаме понятието за модел за внедряване, като едно от основополагащите при взимане на решения, свързани с доставката. Ще се запознаем и с идеята за канал за доставка, полагайки основите за процесуално-издържан подход към предоставянето на софтуер, полагайки рамките, които са ключови за една успешна организация, особено когато е с по-големи размери. И за финал ще разгледаме в детайли трите основни модели на внедряване, като ще се запознаем с характеристиките на всеки от тях.

2 Модел на внедряване

Моделът на внедряване е една от фундаменталните характеристики що се отнася до аспектите, свързани с доставката. Той предопределя огромна част от решенията, които можем да вземем, каналите за доставка (за тях след малко), регулаторните норми и стандартите, с които има нужда да се съобразим и други.

Моделът на внедряване се отнася до различните стратегии и подходи, които организации използват, за да интегрират и внедрят нови софтуерни продукти в техните работни процеси и инфраструктура. Това е план или подход, който организациите избират, за да предоставят софтуерни решение вътрешно в организацията си и/или за своите клиенти, като вземат под внимание фактори като сигурност, контрол, гъвкавост, разходи и удобство.

Внедряването на софтуер може да бъде осъществено чрез различни модели, като "on-premise", "cloud" или "mobile" (които ние ще разглеждаме като нашите основни модели), както и други специфични подходи и комбинации от подходи според конкретните изисквания и цели на организацията. Всеки модел има свои предимства и предизвикателства и може да бъде избран в зависимост от конкретните нужди и ресурси на организацията.

Нека сега се запознаем с естеството на всеки от основните ни модели. Детайлите за тях ще последват по - късно в тази тема.

- **On-premise (На място):** При този модел софтуерът се инсталира и изпълнява върху хардуера и инфраструктурата на самата организация (внедряване за вътрешни нужди) или на клиента (при внедряване с цел външна употреба). Това означава, че компанията / клиента трябва да поддържа собствена инфраструктура, като сървъри и мрежови ресурси. Този модел може да предостави по-голяма контрол и сигурност, но идва с по-големи разходи за инфраструктура и поддръжка. On-premise също би бил и софтуер, инсталиран на потребителските машини.
- **Cloud (Облак):** При този модел софтуерът се доставя чрез интернет на инфраструктура, предоставена от облачни услуги като Amazon Web Services (AWS), Microsoft Azure или Google Cloud Platform. Тук организацията не се занимава с физическа инфраструктура, а заплаща за използването на облачни ресурси според нуждите си. Това позволява гъвкавост, скалируемост и по-ниски начални разходи. Потребителите на такъв тип софтуер го достъпват отдалечено, без необходимост да заделят собствен хардуерен ресурс.
- **Mobile (Мобилен):** Този модел се отнася до внедряване на софтуерни продукти, които се инсталират и изпълняват на мобилни устройства като смартфони и таблети. Този вид внедряване е насочен към потребители, които използват приложенията директно на своите мобилни устройства. Може да се разглежда като частен случай на on-premise, но поради спецификите му го обособяваме като самостоятелен модел.

Изборът на модел на внедряване влияе на начина, по който продуктът се доставя, поддържа и използва, както и на потребителското изживяване. Организациите трябва да анализират подробно характеристиките на всеки модел и да изберат този, който най-добре отговаря на техните специфични цели и изисквания в контекста на конкретен продукт.

Важно е да се отбележи, че е често срещано явление отделни части от даден продукт да имат различен модел на внедряване от останалите части. Например, можем да имаме уеб-базирана услуга за обработка на изображения. Тя е на наша инфраструктура и ние си я поддържаме. Потребителите само обменят данни с нея чрез Интернет и съответните протоколи. Това би бил класически облачен модел за внедряване. Но може да искаме да направим конзолен клиент (command line interface / CLI), за да улесним работата с услугата ни. Самото конзолно приложение се

инсталира от потребителите на техните машини. Правейки този компонент от целия ни продукт пример за on-premise модел.

3 Канал за доставка

Каналите за доставка са важен компонент в цялостния процес на доставка на софтуер. Те са тясно свързани с модела на внедряване тъй като не всеки канал е подходящ за всеки модел. Каналите са със съществен принос за систематичното и регулярно доставяне на софтуер в съответствие с определени правила и предпазването от грешки. Те са особено важни и от гледна точка процеси на продажба и комерсиализация (ще говорим за комерсиализация към края на курса).

Реално те представляват ясно дефиниран процес, който ни казва, при определени характеристики на софтуера и обстоятелствата около доставката, какви са стъпките, през които да се мине, за да може продуктът да достигне до клиента и/или потребителя. Следните аспекти бива да бъдат засегнати при дефинирането на канал за доставка:

- Модели на внедряване, за които е подходящ
- Ниво на развитие на продукта (алфа, бета версии, general availability и други)
- Начин на предоставяне (стандартен, early adopter, OEM, pilot, специализирана разработка за клиент / група клиенти и други)
- Комерсиализация (пробен период (trial), безплатно, регулярно плащане (subscription), еднократно плащане (one time purchase), плащане на база употреба (pay-as-you-go))
- Целеви групи
- Място, където продукта е достъпен (URL / web portal, app store)
- Регламенти за ползване (terms & conditions)
- Тип на съдържанието (patch, hotfix, инсталация, нова версия)
- Изисквания и правила, на които продукта да отговаря (лиценз, съответствие с регулации и стандарти и много други)
- Роли и отговорности
- Стъпки на процеса

Разбира се, всяка фирма запазва правото си да използва своя терминология, да комбинира по различен начин и да разширява представеното в този списък.

С цел предпазване на производителя на софтуера и на клиентите от нежелани или непредвидени обстоятелства, софтуерните продукти и услуги следва да се доставят чрез ясно дефиниран за това канал, одобрен от ръководството и изготвен от съответните специалисти. Така от една страна подsigуряваме, че клиента получава това, което се очаква според договорните отношения. От друга страна гарантира контрол върху това какво, кога и към кой се доставя. Неоторизирана доставка, извън поставените рамки, излага на риск фирмата от към легални и финансови проблеми. Такива могат да бъдат:

- Проблематични задължения за поддръжка
- Нарушения на режимите за експортен и импортен контрол (ще ги разгледаме по - внимателно по - късно в курса)
- Нарушения свързани с лицензите при използването на софтуер с отворен код и друг софтуер
- Нарушения на вътрешнофирмените правила и изисквания
- Проблеми с доказването на интелектуална собственост
- Нарушения в използването на наименования и търговски марки
- Проблеми при отчитането на приходи (поради частична доставка, например)
- Нарушения на регулаторни изисквания и изисквания от стандарти
- Липса на ясни процеси за спиране или оттегляне на доставения софтуер

4 On-premise софтуер

On-premise софтуерът, както самото име подсказва, се инсталира и използва непосредствено на сървъри в инфраструктурата на организацията или на самите машини (персонални компютри / лаптопи) на потребителите, вместо да се достъпва чрез интернет през облачна платформа. Това означава, че при on-premise софтуера всички компоненти, данни и процеси се съхраняват и управляват локално.

Защо да разглеждаме on-premise софтуера в днешно време? Преди всичко, докато облачните технологии стават все по-популярни, все още има фирми, които предпочитат on-premise софтуер по различни причини. Например, организации, които работят със силно регулирани данни или имат изключително високи изисквания към сигурността, често избират on-premise модела, за да запазят пълен контрол над информацията си. В същото време, съществуват и фирми, които предпочитат on-premise софтуера поради специфичните си бизнес нужди, гъвкавост при настройка и възможността за персонализация на софтуера според собствените си изисквания. Не е за подценяване и историческия фактор. On-premise софтуерът дълги години беше стандартът. Фирми с по-дълга история имат значителна съществуваща инвестиция в on-premise инсталации, съответно биха предпочели да продължат да работят с тях, вместо да започнат дълга и времеемка (а и скъпа) миграция към облачните технологии.

Ще разгледаме различните аспекти на доставката на on-premise софтуер, както и предимствата и недостатъците, свързани с този модел.

4.1 Предимства

4.1.1 Контрол и сигурност

Един от основните аргументи в подкрепа на on-premise софтуера е контролът и сигурността, които той предоставя на организацията. Когато инсталирате софтуер на собствените си сървъри и инфраструктура, вие имате пълен контрол над тях и над данните си. Този контрол ви позволява да прилагате специфични политики за сигурност, да наблюдавате достъпа и да управлявате съхранението на данни така, както е необходимо за конкретната ви ситуация. Това прави on-premise софтуера подходящ за фирми, които работят с чувствителна информация и подлежат на строги регулаторни изисквания.

4.1.2 Предвидимост на натоварването

Контролът на достъпа и ексклузивното ползване на инсталацията на софтуера спомага за гарантиране на консистентно поведение и постоянна производителност на софтуера, без значение от външна за нас и нашата организация употреба на същия софтуер (тъй като те използват собствени инсталации). Това не е непременно непреодолим проблем при облачния софтуер, но при on-premise софтуера е нещо, което ни идва на готово.

4.1.3 Конфигуриране

Освен това, on-premise софтуер предоставя голяма гъвкавост при конфигурацията. Вие може да го персонализирате според нуждите на вашата организация и да го настроите точно така, както е необходимо за целите ви. Този контрол ви позволява да създадете уникални решения, които да отговарят на вашите бизнес изисквания. Например, ако вашата фирма изисква специфични интеграции със съществуващи системи или има специални изисквания към процесите си, on-premise софтуерът може да бъде персонализиран така, че да отговори на тези изисквания. Тези изисквания може да произлизат и от нормативната уредба, според която оперираме.

4.2 Недостатъци

Със сигурност, on-premise софтуер има своите предимства, но не можем да пренебрегнем и недостатъците му.

4.2.1 Разходи за оборудване и поддръжка

Един от основните недостатъци е свързан със значителните разходи за оборудване и поддръжка. При инсталирането на on-premise софтуер, вие трябва да закупите и поддържате сървъри, мрежово

оборудване и други ресурси, което може да бъде скъпо както в началото, така и по време на експлоатацията. Тези разходи включват както стойността на хардуера, така и разходите за електроенергия, охлаждане, физическа охрана и обслужване. И не на последно място, необходимо е да осигурим и специалистите, които да внедрят и поддържат софтуера.

4.2.2 Скалируемост

Освен това, on-premise софтуерът обикновено предоставя ограничена скалируемост. Този тип софтуер е свързан с фиксирани ресурси, които не могат да бъдат лесно разширени при нужда. Това може да представлява предизвикателство при растеж на бизнеса или при временни пикове на натоварване, когато е необходимо повече изчислителна мощ и ресурси. Инсталирането и конфигурирането на допълнително оборудване може да отнеме време и да бъде скъпо.

4.3 Процес на доставка

След като разгледахме предимствата и недостатъците на on-premise софтуера, е време да се фокусираме върху процеса на доставка на този вид софтуер. Този процес е от съществено значение, тъй като включва редица стъпки, които организацията трябва да следва, за да осигури успешната инсталация и ефективното функциониране на софтуера.

Като разглеждаме процесуалните стъпки свързани с това софтуера ви да може да се използва от аудиторията, за която е предназначен, ние трябва да се фокусираме върху двата основни вида заинтересовани лица и техните гледни точки. Това са именно доставчика и консуматора на софтуера. Доставчика е производителя на софтуер, който има ангажимента да го предостави в съответствие със всички правила и изисквания. Консуматора пък е този, при който ще се внедри и ще се използва, отново съобразено с изискванията, правилата и регулациите, валидни за ситуацията. Гледната точка на консуматора е много важна и за доставчика, защото в общия случай стъпките на консуматора зависят от и се базират на информация, предоставена от доставчика.

Важно е да се отбележи че в зависимост от процеса на разработка някои стъпки може да са циклични, паралелни и може да има и допълнителни изменения, но на концептуално ниво разглеждаме основните дейности.

4.3.1 Гледна точка на доставчика

Доставчика (производителя) на софтуер е организацията, която го разработва. Тази организация и необходимите действия, които тя трябва да извърши е и основния фокус на курса ни. В нея е основната отговорност за успешното внедряване на софтуера и пазарната му реализация.

4.3.1.1 Планиране и разработка Началните етапи на жизнения цикъл на софтуера не са много специфични спрямо модела на внедряване, но пък са критични за това да имаме софтуерен продукт, който да предоставим. Единствено при тестването и работата с изисквания модела на внедряване е по-силно изразен. Но действително, процеса по доставка е съпътстващ през всичките начални етапи, които разгледахме, когато си говорихме за жизнения цикъл на софтуера. Те създават продукта, който ще доставим. А процеса по доставка следи за спазването на наложените изисквания за успешна реализация.

4.3.1.2 Избор на канал за доставка На база спецификата на софтуерния продукт, следва да се избере точния канал за доставка и да се задействат подготвителните фази на процеса на избрания канал. Трябва да се подсигури, че продукта ни отговаря на всички изисквания на канала.

4.3.1.3 Подготовка за физическата доставка Производителя следва да подготви всички артефакти, които трябва да бъдат предоставени. Това включва, както изпълнимите файлове, които са самия софтуер, така и допълнителни софтуерни елементи като библиотеки, скриптове, спомагателни програми, инсталатори. Дигиталното подписване, криптиране, определянето на лицензирането и внедряването на артефактите, които следват от това (лицензионни файлове) и други такива операции се включват в тези процеси. Също така не трябва да се пренебрегва и документацията. Инструкциите за инсталация, конфигурация и внедряване са критични и трябва да са добре изработени и да обхващат максимално много релевантна информация.

4.3.1.4 Физическата доставка Задействане на процеса според избрания канал за доставка. Така по регламентирания от канала начин, клиенти и/или потребители получават достъп до артефактите, подготвени на предходната стъпка.

4.3.1.5 Поддръжка и актуализации Първичната доставка не изчерпва ангажиментите към продукта. Доставчика следва да предостави поддръжка, консултирайки клиентите, анализирайки проблеми и предоставяйки поправки (функционални, свързани със сигурността, свързани с производителността и други), актуализации. Това също следва да се прави чрез регламентирани канали за доставка. Не е задължително да е чрез първоначално използвания, нито е задължително всеки път да се ползва един и същ. Комуникацията относно тези актуализации следва да е ясна и съпътствана с всички необходими артефакти.

4.3.2 Гледна точка на консуматора

Консуматора е клиента и/или потребителя на нашия софтуер. Отговорен е за внедряването на софтуера. Съответно получава достъп до необходимите инструкции за внедряване и инструментите (като инсталатори), които са предоставени в съответствие с използвания канал за доставка. Именно начина по - който ресурсите достигат до консуматора е един от основните аспекти, изяснени чрез канала за доставка. Консуматора силно зависи от решенията и предоставената информация от страна на доставчика. Ето и основните стъпки, през които се преминава, за да се постигне успешно внедряване.

4.3.2.1 Планиране и изисквания Като начало трябва да се определи какво точно се изисква от софтуера и как ще бъде интегриран във фирмената инфраструктура (или на съответните персонални машини). Ключови въпроси, които трябва да бъдат отговорени, включват:

- Какъв вид софтуер ви е необходим и какви функционалности трябва да включва?
- Колко потребители ще го използват и какви са техните нужди?
- Какъв хардуер и операционна система са необходими за инсталацията?
- Каква сигурност и контрол върху данните се изискват?
- Какви са бюджетните ограничения и сроковете за доставка?

Тази фаза е от решаващо значение, тъй като правилното определение на изискванията ще помогне да се избере правилния софтуер и да се определят ресурсите, които ще бъдат необходими за внедряването. Също така ще са определящи и при конфигуриране в следващите етапи.

Реално на тази стъпка се избира модела на внедряване, който ще се следва. Следователно тя присъства при всеки от моделите.

Работата на доставчика в тази фаза е ключова. Неговата цел трябва да бъде да направи продукта достатъчно привлекателен, за да бъде избран.

4.3.2.2 Придобиване на софтуера При избран модел на внедряване "on premise" и избран конкретен продукт, следва консуматора да се сдобие с него. Било то чрез най-обикновено изтегляне от Интернет на безплатен софтуер или пък чрез по-сложен процес на договори, закупуване на лицензи и други елементи, регламентиращи получаването и използването на продукта. В крайна сметка резултата е наличието на инструменти, инсталатори, документация и всичко друго необходимо за успешното внедряване.

4.3.2.3 Подготовка на инфраструктурата Следва да се определи как софтуера ще се внедри в инфраструктурата. Използвайки документацията и препоръките/изискванията за хардуер, технологии и други, се взимат съответните решения и се изгражда план за внедряване. Тук трябва вземем решение относно сървърите, мрежовото оборудване и всички други компоненти, които ще използваме за инсталацията на софтуера. Фактори, които трябва да се вземат предвид, включват:

- Производителността на хардуера, която ще бъде необходима за системата.
- Съответствието със специфичните изисквания на софтуера.

- Сигурността и надеждността на хардуера.
- Бъдещите нужди от разширение и скалируемост.
- Необходимост от виртуални машини/контейнери
- Необходими операционни системи и друг софтуер

Правилните решения относно инфраструктурата са от съществено значение, тъй като те ще определят производителността и стабилността на софтуера в продължение на целия му жизнен цикъл.

4.3.2.4 Внедряване Следва същинското внедряване на софтуерния продукт. Според изградения план за инфраструктурата, започва:

- Включването на сървъри, виртуални машини, клъстери и други.
- Конфигуриране на мрежови устройства
- Инсталация на операционни системи
- Инсталация на базов софтуер и зависимости
- Инсталация на самия софтуерен продукт
- Конфигурация и настройка на софтуера според изискванията на организацията. Това може да включва конфигуриране на бази данни, настройка на потребителски роли и права за достъп, както и други настройки на самия софтуер.

4.3.2.5 Тестване и оптимизация Успешното внедряване на техническо ниво може и да завършва щом софтуера е работещ и готов за употреба. Но този софтуер не носи стойност ако не удовлетворява нуждите, за които е предвиден. Затова е важно да се валидира инсталацията.

- **Тестове на функционалността:** тестване на различните функционалности на софтуера, за да се провери дали те работят както трябва. Разглежда се всяка част от приложението и се проверява, че няма грешки или проблеми.
- **Тестове на производителността:** измерване производителността на софтуера при натоварване и определяне дали сървърите могат да се справят с него при текущите конфигурации. Ако се наблюдават забавяния или изключения, се разглеждат опции за оптимизация.
- **Тестове за сигурността:** идентифициране и коригиране на потенциални рискове и уязвимости в системата.

4.3.2.6 Поддръжка и актуализации Софтуерът може да е вече в експлоатация, но това не означава, че няма повече дейности. Необходимо е да се осигури, че софтуера се наблюдава, за да може да се реагира при възникване на ситуации като надвишено натоварване, проблеми с инфраструктурата и други. Също така трябва регулярно да се внедряват актуализации и корекции (patches), предоставени от доставчика. Това е важно, защото те съдържат поправки на различни проблеми, които може да са от сериозно значение за консуматора на продукта.

4.4 Сигурност

Сигурността при on-premise софтуера има свои специфични изисквания и предизвикателства, които трябва да бъдат взети предвид:

- **Пълнен контрол:** Отличителна черта на on-premise софтуера е пълният контрол върху инфраструктурата и данните. Това предоставя възможност за персонализирани настройки за сигурността.
- **Физическа сигурност:** Фирмите трябва да осигурят физическа сигурност на сървърните си помещения и оборудване за предотвратяване на неоторизиран достъп.
- **Интелигентен мониторинг:** Системите за мониторинг и реакция при инциденти трябва да бъдат внимателно конфигурирани, за да засичат потенциални заплахи.

- **Редовни корекции и обновления:** Фирмите трябва активно да следят за поправки, свързани със сигурността, и да ги прилагат, за да се предотвратят уязвимости.
- **Защита на данни и криптиране:** Контролът на данните позволява използването на надеждни методи за защита и криптиране на информацията.
- **Инцидентен план:** Изграждането на инцидентен план и обучението на персонала по него е от съществено значение за бърза реакция при инциденти.

Сигурността при on-premise софтуера изисква навлизане в детайли за физическата инфраструктура и активното управление на сигурността, което може да бъде по-сложно, но предоставя пълен контрол и гъвкавост на организацията.

4.5 Скалируемост и оптимизация

Скалируемостта и оптимизацията представляват ключови фактори в успешното управление на on-premise софтуерни системи. В тази секция ще разгледаме стратегии и методи, които фирмите могат да приложат, за да разширят и подобрят своите on-premise решения. Изключително важно е производителите на on-premise софтуер да предоставят необходимите указания, които да спомогнат консуматора да изгради своя план.

4.5.1 Скалиране

Скалирането на on-premise решението е необходимо, когато се увеличава трафика и натоварването. Ето някои ключови аспекти:

- **Хоризонтално скалиране:** Този метод включва добавянето на допълнителни сървъри или ресурси на вече наличната инфраструктура. Той позволява равномерно разпределение на трафика и по-голяма надеждност.
- **Вертикално скалиране:** Този подход включва увеличаване на мощността на сървърите или инфраструктурата чрез добавяне на по-мощни компоненти или ресурси към съществуващите сървъри. Той може да подобри производителността, но се използва по-рядко от хоризонталното скалиране.
- **Използване на виртуализация и контейнери:** Виртуализацията и контейнеризацията позволяват по-ефективно управление на ресурсите, като позволяват на един сървър да работят множество виртуални машини или контейнери. Това подпомага оптималното използване на хардуера и ресурсите.

4.5.2 Оптимизация на ресурсите

Оптимизацията на ресурсите е ключов аспект в on-premise решенията, тъй като тя може да намали излишните разходи и да подобри ефективността. Ето какво следва да се вземе предвид:

- **Оптимизация на оборудването:** Разглеждат се възможности за актуализация на хардуера или преминаване към по-модерни технологии, които да намалят разходите и да увеличат производителността.
- **Ефективно управление на енергопотреблението:** Разработване на практики за намаляване на консумацията на електроенергия и поддържане на екологичността на дейността.
- **Използване на облачни и виртуални решения:** Разглеждане на възможността да се интегрират облачни решения (създаване на хибриден облак) в инфраструктура.

4.6 Актуализации

За успешното функциониране на on-premise софтуерните системи е от съществено значение да се поддържат актуални версии. Обновленията на софтуера са ключов момент за подобряване на сигурността, функционалността и ефективността на системата. Доставчика на софтуер следва да разработи обновлението в зависимост от своите планове за развитие на продукта, след което да състави нова версия и да я достави според утвърдените правила. Консуматора следва да

прецени кога, дали и по какъв начин да внедри актуализацията. Обновленията може да съдържат и поправки на дефекти. Обикновено ако такава поправка се прави с нормална версия (а не извънредна такава), то става дума за дефект, който не е толкова критичен.

Обновления може и да се правят в отговор на открити уязвимости или критични дефекти. Те са с много по-кратко време за разработка и не съдържат допълнително съдържание. Доставят се често със специални процедури, които съкращават необходимото време и следва консуматорите да бъдат уведомени специално за тях и рисковете при неприлагане.

5 Cloud софтуер

Cloud софтуер представлява еволюция в доставката на софтуерни приложения и услуги. Той осигурява гъвкавост и удобство във връзка с достъпа, използването и управлението на софтуера, като предоставя тези услуги чрез Интернет, посредством облачна инфраструктура. Този модел на доставка се различава от традиционния софтуер, инсталиран директно на локалните компютри и сървъри, разгледан в предната част.

Моделите на доставка тук са под формата на услуги, давайки на потребителите възможността да използват софтуера и функционалностите му без да се грижат за него. Съответно се характеризира с по-различна комерсиализация и договорни отношения. Ключов елемент е така наречения *service level agreement (SLA)* - писмено споразумение между доставчик на услуги и клиент, което дефинира конкретните условия и стандарти, които трябва да се спазват при предоставянето на услугата. Това включва параметри като ниво на наличност, време за реакция на проблеми и обслужване на клиенти, и се използва за определяне и гарантиране на качеството и надеждността на услугата, както и за установяване на права и отговорности на страните при евентуални нарушения на условията. SLA е важен инструмент за управление на отношенията между доставчици на услуги и техните клиенти, осигурявайки яснота и прозрачност във отношенията им.

Важно е да разберем основните модели за доставка на Cloud софтуер и тяхната същност (списъкът далеч не е изчерпателен):

- **Software as a Service (SaaS):** Този модел предоставя готови софтуерни приложения и услуги чрез Интернет. Потребителите използват SaaS приложенията директно, без необходимостта да се грижат за инсталиране, поддръжка и обновления. Примери за SaaS включват Google Workspace, Microsoft 365 и Figma.
- **Platform as a Service (PaaS):** PaaS предоставя платформа и средства за разработка на софтуер. Това позволява разработчиците да създават и внедрят свои приложения без да се занимават с инфраструктурни детайли като хардуер и операционни системи. Популярни PaaS доставчици включват AWS Elastic Beanstalk и Google App Engine.
- **Infrastructure as a Service (IaaS):** IaaS предоставя виртуални ресурси за сървъри, мрежи и съхранение. Този модел позволява организациите да управляват и конфигурират инфраструктурата си чрез Интернет, елиминирайки нуждата да поддържат свой хардуер. Потребителите имат пълен контрол над операционните системи и приложенията. Примери за IaaS включват Amazon Web Services (AWS), Microsoft Azure и Google Cloud Platform.
- **Function as a Service (FaaS):** FaaS е модел, който позволява на разработчиците да изпълняват индивидуални функции (или микроуслуги) в реално време, без да се грижат за процесите, които изпълняват функцията. Това осигурява еластичност и оптимизация на използваните ресурси. Примери за FaaS включват AWS Lambda и Azure Functions.

Тези "as a Service" модели предлагат различни предимства, включително по-голяма гъвкавост, намалени разходи за инфраструктура и удобство при управлението на софтуера. В следващите раздела на лекцията, ще разгледаме какво прави Cloud софтуера толкова атрактивен и как да го доставяме и управляваме успешно.

5.1 Предимства

Cloud софтуерът представлява революционен начин за предоставяне на софтуерни приложения и услуги. Той предоставя множество предимства, които променят начина, по който организациите разработват и доставят софтуерни продукти. В тази секция ще разгледаме някои от ключовите предимства на Cloud софтуера.

5.1.1 Скорост и гъвкавост в развитието

Едно от големите предимства на Cloud софтуера е неговата способност да ускори и опрости процеса на разработка и доставка на приложения. С облачните инфраструктури и инструменти за разработка, разработчиците могат да създават и тестват нов софтуер по-бързо и по-ефективно. Еластичността на облачните ресурси позволява мигновеното мащабиране на приложенията спрямо нуждите на потребителите, без да се налага инвестиране в нови физически сървъри.

Гъвкавостта на облачния софтуер позволява на компаниите да се адаптират бързо към променящите се изисквания и пазарни условия. Това означава, че организациите могат да бъдат по-конкурентоспособни и да реагират по-бързо на възможности или предизвикателства.

5.1.2 Намалени разходи за оборудване и поддръжка

Едно от ключовите икономически предимства на Cloud софтуера е свързано с намаляването на инвестициите във физическо оборудване и поддръжка. Традиционните методи за разработка изискват значителни финансови ресурси за закупуване и поддръжка на сървъри, мрежово оборудване и центрове за данни (data centers). С облачния софтуер, тези разходи се намаляват, тъй като инфраструктурата се предоставя като услуга от доставчиците.

Поддръжката на хардуера, обновлението и сигурността са задачи, за които се грижи доставчика, освобождавайки организациите от тези задачи. Това прави облачния софтуер достъпен и изгоден както за големи, така и за малки предприятия.

Със своите скорост и икономичност, Cloud софтуерът става ключова част от съвременната софтуерна индустрия и играе важна роля в подобряването на бизнес процесите и конкурентоспособността на организациите.

5.1.3 Скалируемост и гъвкавост в ресурсите

Cloud софтуерът позволява на организациите лесно да мащабират ресурсите спрямо нарастващите нужди. Това означава, че при увеличаване на натоварването или трафика към приложението, ресурсите могат да бъдат увеличени автоматично. Същевременно, при по-ниска натовареност ресурсите могат да бъдат намалени, което спестява разходи.

5.1.4 Глобална достъпност

Облачните решения позволяват на потребителите да имат глобален достъп до приложенията и данните. Това е от особена важност за глобални организации и тези, които целят клиенти в различни части на света. Облачните доставчици разполагат с инфраструктура в различни региони, което осигурява надеждна и бърза достъпност.

5.1.5 Автоматизация и управление на ресурсите

Cloud софтуерът предлага инструменти за автоматизация на различни аспекти от управлението на инфраструктурата и приложенията. Това включва автоматично мащабиране, управление на контейнери и оркестрация на ресурсите. Автоматизацията подобрява ефективността и намалява възможността за човешка грешка. Като е важно да се отбележи, че тук автоматизацията идва в голяма степен на готово, а не е нещо, което е невъзможно при on-premise софтуера.

5.1.6 Бързо въвеждане на нови функции и актуализации

С облачния софтуер, актуализациите и новите функции могат да бъдат внедрявани по-бързо и без значително прекъсване на услугите. Това позволява на организациите да бъдат конкурентоспособни и да реагират на бързопроменящите се изисквания и тенденции на пазара.

5.2 Недостатъци

Въпреки многобройните предимства, които предоставя Cloud софтуерът, съществуват и някои недостатъци, които трябва да бъдат взети предвид при използването му. В тази секция ще разгледаме някои от основните недостатъци на облачния софтуер.

5.2.1 Зависимост от Интернет свързаността

Един от основните недостатъци на Cloud софтуера е зависимостта от стабилна и бърза Интернет свързаност. Ако потребителите нямат достъп до интернет или латентността (закъснението) е висока, достъпът до облачния софтуер може да бъде затруднен или неудовлетворителен. Това може да създаде проблеми, особено при работа в отдалечени или лошо свързани региони.

5.2.2 Въпроси за сигурност и контрол

Сигурността на данните и контролът върху тях са сериозна грижа за много организации при използването на облачен софтуер. При прехвърлянето и съхранението на чувствителна информация в облака, съществува риск от неразрешен достъп, злоупотреба или изтичане на данни. Организацията трябва да предприемат допълнителни мерки за сигурност и да установят строг контрол върху данните си, което може да бъде сложно и скъпо. За намаляване на ефекта от този недостатък е ключова сертификацията по различни стандарти за сигурност.

5.2.3 Сложност на интеграция със съществуващи системи

За много организации преходът към облачен софтуер може да бъде сложен и изисква интеграция със съществуващи системи и приложения. Не всички приложения и данни могат да бъдат лесно мигрирани в облака, и това може да предизвика прекъсване на работния процес и да налага допълнителни разходи за приспособяване.

5.2.4 Ограничения на контрола и настройките

При използването на публични облачни услуги, потребителите често се ограничават във възможността си да настройват инфраструктурата и средата за изпълнение на софтуера. Това ограничение на контрола може да бъде пречка, особено за организации, които изискват висока степен на персонализация и контрол над средата си.

5.2.5 Преносимост на данни

При необходимост от пренос на големи обеми данни между различни облачни доставчици или между облака и локални системи, може да възникнат проблеми със скоростта и разходите. Трансферът на данни през Интернет може да бъде бавен и да предизвика допълнителни разходи за мрежова свързаност.

5.3 Процес на доставка

Доставката на Cloud софтуер е сложен процес, който изисква внимателно планиране, конфигуриране и управление от страна на доставчика. В тази секция ще разгледаме процеса на доставка на Cloud софтуер от гледна точка на доставчика и ще обясним важните етапи и разглеждания.

Важно е да се отбележи че в зависимост от процеса на разработка някои стъпки може да са циклични, паралелни и може да има и допълнителни изменения, но на концептуално ниво разглеждаме основните дейности.

5.3.1 Планиране

Първият и ключов етап в процеса на доставка на Cloud софтуер е планирането и събирането на изискванията на клиента. Доставчикът трябва да разбере нуждите и целите на клиента и да определи какви ресурси и инфраструктура са необходими за доставката на софтуера. Това включва определяне на обема на данните, изискванията за сигурност, производителността и други параметри. Това ще включва и определяне на модела на доставка.

5.3.2 Избор на облачен доставчик и конфигурация

След като са установени изискванията, доставчикът трябва да избере подходящата облачна инфраструктура и да конфигурира средата за доставка на софтуера. Това включва избор на оборудване, мрежови настройки, настройки за сигурност и други конфигурационни детайли. Важно е доставчикът да осигури оптимална среда за работа на софтуера и да гарантира наличието на необходимите

ресурси. Ще се определят и какви от възможностите, предоставени от облачния доставчик ще бъдат използвани.

5.3.3 Разработка

Следва дизайн на решението и разработка. Архитектурата силно се влияе от предходния етап. Съответно следва и да се зачете техническото мнение на архитектите при избора. В процеса на разработка следва да се осигури облачна среда, идентична с продуктивната, с цел използване от разработчиците.

5.3.4 Тестване и оптимизация

Доставчикът трябва да извърши тестване на софтуера, за да се увери, че всичко функционира коректно. Това включва функционални тестове, такива с цел идентифициране на проблеми и оптимизация на производителността и тестове на сигурността. Важно е да се осигури, че софтуерът работи стабилно и отговаря на изискванията на клиента. Това е прието да се прави на предвидена за тестване среда според ясно дефиниран процес.

5.3.5 Избор на канал за доставка

На база спецификата на софтуерния продукт, следва да се избере точния канал за доставка и да се задействат подготвителните фази на процеса на избрания канал. Трябва да се подсигури, че продукта ни отговаря на всички изисквания на канала.

5.3.6 Подготовка за физическа доставка

Следва да се подготвят всички артефакти, които са необходими за внедряването. Това включва както изпълнимите файлове, които са самия софтуер, така и допълнителни софтуерни елементи като библиотеки, скриптове, спомагателни програми, инсталатори. Дигиталното подписване, криптиране, определянето на лицензирането и внедряването на артефактите, които следват от това (лицензионни файлове) и други такива операции се включват в тези процеси. Трябва да се определят конфигурационни параметри спрямо направените тестови и да се подготви автоматизацията за физическата доставка. Еwentуално, в зависимост от правилата на канала, може да се изисква разрешение и/или уведомление.

5.3.7 Физическа доставка

Задействане на процеса според избрания канал за доставка. Така по регламентиран от канала начин, клиента и/или потребителя получава достъп до софтуерния продукт (неговите функционалности и лимитирана възможност за конфигуриране).

Разбира се, има голяма свобода в процеса, дефиниран от канала. Но в общия случай се наблюдава следното:

1. Софтуера (или нова негова версия) се внедрява на тестова среда. Там еwentуално се внедряват и други зависими продукти, за да се наблюдава и интеракцията между тях.
2. Изпълнява се верификация, за да се подсигури желаното ниво на качество.
3. Софтуера (или нова негова версия) се внедрява пилотно на някоя продуктивна среда. Често това се нарича *canary deployment* (ако ви е интересно наименованието, прочетете за използването на канарчета в миньорството). Обикновено тази среда е вътрешно-продуктивна за организацията. Тоест, използва се от самия производител за свои нужди, но не и от клиенти. Така при еwentуални поражения, те ще са локализирани вътре в организацията.
4. Наблюдава се поведението при тази истинска употреба и се анализират потенциални индикации за проблеми.
5. Софтуера (или нова негова версия) се внедрява на останалите среди и става де факто текущата и единствена версия на продукта.

За да се постигне абсолютно равенство между средите е от изключително значение наличието на автоматизация. Така се избягват разминавания и човешки грешки.

5.3.8 Поддръжка и актуализация

Cloud софтуерът не спира да се развива и доставчикът трябва да осигури редовни обновления на софтуера. Тези обновления могат да включват нови функционалности, подобрения на сигурността и поправки на грешки. Доставчикът трябва да следи актуалните тенденции и да осигурява актуализации, приваждайки софтуера в съответствие с новите изисквания на клиента.

5.3.9 Мониторинг и управление на ресурсите

Процесът на доставка не завършва с внедряването на софтуера. Доставчикът трябва да осигури постоянен мониторинг на облачната инфраструктура и приложенията. Този мониторинг помага да се идентифицират проблеми и да се предотвратят инциденти. Управлението на ресурсите включва оптимизация на използваните ресурси, за да се осигури ефективност и икономия на разходи. А при възникване на инциденти, трябва да се реагира според ясно установени процеси и в съответствие с SLA документа.

5.4 Сигурност

5.4.1 Защита на данни и достъп

Облачните услуги обработват големи обеми от данни, включително чувствителна информация за клиенти, транзакции и други. За да се осигури сигурността на данните, е важно да се предприемат следните мерки:

- **Шифриране на данните:** Шифриране както при пренасянето, така и при съхранението на данни в облака. SSL/TLS за мрежовия трафик и шифриране на данните в покой със силни алгоритми.
- **Контрол на достъпа:** Настройване на детайлни права за достъп и автентикация, използвайки двуфакторна идентификация, силни пароли и многофакторни методи за контрол на идентичността на потребителите.
- **Редовни одити и мониторинг:** Следене и записване дейностите на потребителите и системата, за да можете да се забележат неизправности или атаки в реално време.
- **Firewalls и сегментация:** Използване на firewall и мрежова сегментация, за да се ограничи достъпа до облачни ресурси и да се предотврати неоторизиран достъп.

5.4.2 Реакция при инциденти и бекъп решения

Способността да се реагира при инциденти и да се възстанови информацията е от изключително значение:

- **Управление на инциденти:** Необходим е план за управление на инциденти, който да включва мерки за бързо откриване и реагиране при потенциални инциденти, свързани със сигурността.
- **Резервни копия (back-ups):** Редовно правене на back-ups на данните си и тестване на процедурите за възстановяване, за да може да се възстанови оперативността след инцидент.
- **Безопасност при изход:** Планиране как да се осигури безопасността на данните при напускане на облачния доставчик или при прекратяване на услугата.

5.4.3 Обучение и осведоменост

Сигурността на облачния софтуер изисква постоянно обучение и повишена осведоменост сред персонала.

5.5 Скалируемост и оптимизация

Скалируемостта и оптимизацията са ключови аспекти при изграждането и управлението на облачен софтуер. Те позволяват на приложенията и ресурсите да се адаптират към нарастващите нужди и да работят ефективно. Ето някои от важните аспекти на скалируемост и оптимизация:

5.5.1 Скалиране на Cloud решения

Скалирането в облака позволява на софтуерът да увеличи или намали капацитета си според натоварването. Това осигурява следните предимства:

- **Висока наличност:** Скалирането позволява да се осигури висока наличност на услугите, като се гарантира, че има резервни ресурси, готови да поемат трафика при натоварване или инциденти.
- **Избягване на претоварване:** Автоматичното скалиране предотвратява претоварването на ресурсите и осигурява непрекъснато качество на услугата, дори при внезапни натоварвания.

Скалирането е бързо тъй като използва облачната инфраструктура и не изисква тепърва да се пускат сървъри и други. Моментално може да се резервира допълнителен ресурс и да бъде вкаран в употреба.

5.5.2 Оптимизация на ресурсите в облака

Оптимизацията на ресурсите е процесът на настройка и управление на облачните ресурси, за да се осигури максимална ефективност и минимални разходи. Някои от основните стратегии включват:

- **Мониторинг и анализ:** Редовен мониторинг на използването на ресурсите и анализ на данните, за да се идентифицират възможни точки за оптимизация.
- **Автоматизация:** Използване на автоматизирани инструменти за управление на ресурсите и процесите за автоматично скалиране и настройка.
- **Използване на резервации:** Използване на резервации на ресурси, когато се знае, че ще има постоянна нужда, за да се спестят разходи.

5.5.3 Балансиране на натоварването

Балансирането на натоварването (load balancing) е важен компонент на скалируемостта и оптимизацията. Този процес разпределя трафика между множество ресурси или сървъри, за да се постигне следното:

- **Равномерно натоварване:** Балансьорът на натоварването гарантира, че всички ресурси са равномерно натоварени и че няма излишно натоварване на някои от тях.
- **Висока наличност:** В случай на отказ на един ресурс, баланссьорът автоматично пренасочва трафика към други ресурси, осигурявайки непрекъснатост на услугата.
- **Подобрена производителност:** Балансьорът може да подобри производителността, като разпределя трафика спрямо текущите натоварвания и способностите на ресурсите.

5.6 Актуализации

Актуализациите на софтуера са от съществено значение за сигурността, функционалността и производителността на облачните решения. Също така и за продължителното удовлетворяване на клиентските нужди. В тази секция ще разгледаме важността на редовните актуализации и как да се управляват.

5.6.1 Защо са необходими редовни актуализации?

Обновленията на облачния софтуер служат за няколко основни цели:

- **Сигурност:** Обновленията често съдържат поправки на уязвимости, които могат да бъдат използвани от злонамерени страни. Редовните обновления помагат да се поддържа сигурността на данните и инфраструктурата.
- **Поправяне на грешки:** Обновленията често включват корекции на грешки и проблеми, които могат да доведат до нестабилност на системата, некоректно поведение или дори до нейното сриване.

- **Добавяне на нови функционалности:** Новите версии на софтуера често внасят подобрения и нови функционалности, които могат да повишат ефективността и производителността на облачните приложения и да отключат нови сценарии и възможности.
- **Съвместимост:** Редовните обновления могат да поддържат съвместимост с актуални операционни системи, браузъри и други зависимости на софтуера.

5.6.2 Управление на актуализациите

Управлението на актуализациите изисква грижливо планиране и изпълнение, както следва:

- **Планиране:** Определяне на план, който да включва редовни проверки и актуализации на облачния софтуер.
- **Тестове:** Преди да се направи актуализация в продуктивна среда се извършват подробни тестове в тестова среда, за да се подсигури, че актуализацията няма да предизвика нежелани проблеми.
- **Автоматизация:** Използване на автоматизирани инструменти и процеси, за да се облекчи и оптимизира управлението на актуализациите.
- **Следене и възможност за възстановяване на предишното състояние:** Прилагането на актуализации следва да се наблюдава внимателно и да се отстраняват проблеми при самото прилагане. Ако пък се окаже, че актуализацията създава проблеми те трябва или да се разрешат бързо, или да се премахне актуализацията, възстановявайки състоянието преди него.

5.6.3 Актуализации от облачния доставчик

При използването на облачни среди е важно да се следи за актуализации и от страна на облачния доставчик. Актуализациите в облачната инфраструктура могат да включват множество важни аспекти, които нашия софтуер може да има нужда да вземе предвид.

6 Mobile софтуер

Мобилният софтуер е специално разработен за използване на мобилни устройства, като смартфони и таблети. Този вид софтуер има изключително важно значение в съвременния свят, поради своето влияние върху начина, по който потребителите се свързват с информацията и взаимодействат с технологичните решения.

Същността на мобилния софтуер е във възможността да предоставя персонализиран и бърз достъп до разнообразни услуги и данни. Тези приложения са специално проектирани, за да работят на устройства с ограничени ресурси и екранни размери, но и със специфични хардуерни възможности и свързаност. Те предоставят удобство и леснота на потребителите, като им позволяват да изпълняват задачи, без значение къде се намират.

Мобилният софтуер съчетава хардуерните и софтуерните възможности на устройствата, включително сензори, камери и геолокация, за да предостави персонализирано и съобразено с контекста изживяване. Това позволява разработчиците да създават иновативни приложения, които обогатяват и подобряват живота на потребителите.

Същността на доставката на мобилен софтуер частично наподобява тази при on-premise софтуера. Все пак има своите специфики и разлики, особено що се отнася до начина за дистрибутиране и внедряване. Често mobile софтуера взаимодейства с други (най-често cloud) решения, за да предостави цялата функционалност на своите потребители.

6.1 Предимства

Мобилният софтуер има ключова роля в трансформацията на социалните, комуникационните и бизнес процеси, като предоставя редица предимства, които се отразяват върху начина, по който потребителите и организациите функционират. Някои от най-съществените предимства на мобилния софтуер са:

- **Портативност:** Мобилните устройства са преносими и малки, позволявайки на потребителите да имат достъп до информация и услуги по всяко време и навсякъде. Мобилното устройство е неизменна част от живота на съвременния човек, затова и мобилния софтуер е по-лесно достъпен в ежедневието.
- **Персонализация:** Мобилният софтуер може да използва данни от сензори и геолокация, за да предостави персонализирани и контекстуални услуги. Например, приложенията за навигация могат да предложат маршрут, съобразен с текущото местоположение и трафика.
- **Интерактивност и ефективност:** Сензорите на мобилните устройства, като сензори за допир и акселерометри, предоставят интуитивни начини за взаимодействие с приложенията. Това увеличава ефективността и удовлетворението на потребителите.
- **Подобрени възможности за бизнеса:** Мобилният софтуер предоставя нови възможности за бизнес сектора, включително мобилни плащания, маркетинг на място и събиране на данни от потребителите. Тези инструменти подпомагат иновациите и конкурентоспособността на компаниите.

Съчетавайки тези предимства, мобилният софтуер дава възможност за трансформация на начина, по който обществото и бизнесите функционират.

6.2 Недостатъци

Въпреки множеството предимства, мобилният софтуер също така се сблъсква с редица недостатъци и предизвикателства, които трябва да бъдат взети предвид при разработката и използването му. Някои от основните недостатъци на мобилния софтуер включват:

- **Ограничени ресурси:** Мобилните устройства разполагат с ограничени ресурси като процесорна мощност, RAM и дисково пространство. Това може да ограничи функционалността и производителността на някои приложения.
- **Разнообразие от устройства и ОС:** Съществува голямо разнообразие от мобилни устройства (с различни сензори и размери на екрана) и операционни системи (ОС), които разработчиците трябва да поддържат. Това води до сложност при тестване и оптимизация на приложенията за различни платформи.
- **Ограничена батерия:** Батерията на мобилните устройства е ограничена. Приложения, които използват много енергия, могат да доведат до по-бързо спадане на нивото на заряд и да доведат до недоволство сред потребителите.
- **Сигурност:** Мобилните устройства са подложени на различни видове заплахи за сигурността, включително зловреден софтуер, фишинг и уязвимости на приложенията. Това изисква постоянна грижа за сигурността на мобилните приложения.
- **Съвместимост и обновления:** Обновленията на операционните системи и хардуера могат да доведат до проблеми със съвместимостта за стари приложения. Разработчиците трябва да поддържат актуализации, за да осигурят работоспособността на приложенията.
- **Ограничен офлайн достъп:** По-голямата част от мобилните приложения изискват Интернет връзка за пълноценно функциониране. Този ограничен офлайн достъп може да създаде неудобство при липса на Интернет свързаност.

Изследването и решаването на тези недостатъци са ключови за разработка на успешни мобилни приложения и за осигуряване на удовлетворението на потребителите.

6.3 Процес на доставка

Процесът на доставка на мобилен софтуер е сложен и изисква внимателно планиране и изпълнение. Включва няколко ключови стъпки и фактори, които са от съществено значение за успешната доставка на мобилните приложения.

Важно е да се отбележи че в зависимост от процеса на разработка някои стъпки може да са циклични, паралелни и може да има и допълнителни изменения, но на концептуално ниво разглеждаме основните дейности.

6.3.1 Планиране

Отново, както и при другите модели за внедряване, първият и ключов етап в процеса на доставка на мобилен софтуер е планирането и събирането на изискванията на клиента. Доставчикът трябва да разбере нуждите и целите на клиента и да определи какви ресурси и инфраструктура са необходими за доставката на софтуера. Това включва определяне на обема на данните, изискванията за сигурност, производителността и други параметри. Това ще включва и определяне на модела на доставка.

6.3.2 Разработка

Следва дизайн на решението и разработката му. В общия случай по време на разработката се използват емулятори, симулиращи мобилно устройство.

6.3.3 Тестване и оптимизация

Трябва да се извърши тестване на софтуера, за да се подsigури коректността на функционалността на системата. Това включва функционални тестове, такива с цел идентифициране на проблеми и оптимизация на производителността и тестове на сигурността. Приложението трябва да бъде интегрирано с мобилната операционна система и да бъде оптимизирано за използване на ресурси и производителност. Това включва съвместимост с различни версии на операционната система и устройства.

6.3.4 Избор на канал за доставка

На база спецификата на софтуерния продукт, следва да се избере точния канал за доставка и да се задействат подготвителните фази на процеса на избрания канал. Трябва да се подsigури, че продукта ни отговаря на всички изисквания на канала.

6.3.5 Подготовка за физическа доставка

Следва да се подготвят всички артефакти, които са необходими за внедряването. При мобилния софтуер често има ясен и унифициран формат на необходимите артефакти в зависимост от операционната система. Евентуално, в зависимост от правилата на канала, може да се изисква разрешение и/или уведомление.

6.3.6 Физическа доставка

Задействане на процеса според избрания канал за доставка. Така по регламентиран от канала начин, клиента и/или потребителя получава достъп до софтуерния продукт (неговите функционалности и лимитирана възможност за конфигуриране).

Обикновено приложението ще бъде публикувано в официалните магазини за приложения като Google Play Store и Apple App Store.

6.3.7 Поддръжка и актуализация

Процесът на доставка на мобилен софтуер не приключва с публикацията. Разработчиците трябва да предоставят поддръжка на приложението и да издават редовни обновления, включително актуализации за сигурността и нови функции.

6.4 Сигурност

Сигурността в мобилния софтуер е специфична поради характеристиките на мобилните устройства. Ще се фокусираме върху някои от аспектите на сигурността:

- **Управление на приложенията:** Мобилните операционни системи имат строг контрол върху това как приложенията могат да взаимодействат с устройството и данните. Това включва разрешения за достъп до камера, микрофон, GPS и други чувствителни ресурси. Разработчиците трябва да използват тези разрешения разумно и да осигурят, че приложенията са минимално привилегирани.

- **Специфични атаки:** Мобилните устройства са подложени на специфични видове атаки, като например атаки със зловреден софтуер, фишинг и дори атаки върху безжични мрежи. Сигурността трябва да се фокусира върху предотвратяването и разкриването на тези атаки.
- **Загуба на устройството:** Загубата или кражбата на мобилното устройство може да представлява сериозна заплаха за сигурността. По тази причина, мобилните операционни системи предоставят инструменти за дистанционно изтриване на данни, заключване на устройството и проследяване на неговото местоположение. Но и трябва да се вземат мерки за предотвратяване на злоупотреби преди потребителя да може да реагира.
- **Сигурност на данни при движение:** Приложенията за мобилни устройства често изпращат данни по мрежата. За да се осигури техните сигурност и поверителност, трябва да се използват протоколи за шифроване на данните, като например SSL/TLS, дори при използване на сигурни Wi-Fi мрежи.
- **Сигурност на хранилището на данни:** Данните, съхранявани на мобилното устройство, трябва да бъдат шифровани. Това включва данни във вътрешната памет, на SD карти и съхранени в облака.
- **Сигурност на обновленията:** Редовните актуализации на операционната система и приложенията са от съществено значение за сигурността. Разработчиците трябва да се грижат за актуализациите и да ги предоставят на потребителите, за да се отстранят уязвимости и да се подобри сигурността.

6.5 Скалируемост и оптимизация

При мобилните приложения имаме вградена скалируемост, идваща по дефиниция. Това се дължи на факта, че всеки потребител има своя инсталация. В този ред на мисли не е нужно да говорим за скалируемост.

Оптимизацията обаче е ключова. Ограниченият ресурс, желанието да не се злоупотребява със степента на заряд на батерията, желанието да не се използва излишна памет и да не се пречи на други приложения. Това са важни аспекти, които трябва да се вземат предвид и да се правят постоянни оптимизации, но без това да има негативно влияние върху софтуера и неговото функциониране.

6.6 Актуализации

Актуализациите на мобилния софтуер са важен аспект от жизнения цикъл на всяко мобилно приложение. Те не само добавят нови функционалности и подобряват производителността, но и са от критично значение за сигурността на приложението и устройството. Някои от важните аспекти на обновленията на мобилния софтуер са:

- **Редовни обновления:** Редовните актуализации на операционната система и приложенията са от съществено значение за сигурността на мобилния софтуер. Те включват поправки на уязвимости и подобрения на сигурността, които са критични за защитата на данните и личната информация на потребителите. Също така включват и нови функционалности и оптимизации.
- **Източник на обновления:** Потребителите трябва да изтеглят актуализации само от официални и надеждни източници, като Google Play Store за Android и Apple App Store за iOS. Инсталирането на софтуер от недоверени източници може да представлява риск за сигурността. Разбира се, източника в крайна сметка е регламентиран от канала за доставка и при специфични случаи може да не бъде някой от стандартните.
- **Автоматични обновления:** Предоставянето на възможност за автоматични актуализации е добра практика. Това позволява на приложенията и операционната система да се актуализират автоматично, когато нови версии станат налични. Това не само подобрява сигурността, но и осигурява, че потребителите използват най-новите и оптимизирани версии на софтуера.

7 Заключение

Разбирането за видовете софтуер, моделите на внедряване и ясно дефинираните канали за доставка са ключови за успешната и регулярна доставка на софтуери продукти в дълъг период от време. Различните продукти имат своите специфики и нужди, затова е много важно те да бъдат добре разбрани и да се взимат правилните решения, касаещи доставката.