# Глава 1. Проблеми на информационното осигуряване при управление на поръчките от клиенти

## 1.1. Същност и принципи на информационните системи, поддържащи дейността на производствено предприятие

### 1.1.1. Характеристика на ERP системите

ERP информационните системи позволяват ефективното планиране на дейността на предприятие, в т.ч. разходи за обновяване на оборудването и инвестициите в производството на нови изделия (Банабакова, В. К, 2019). Те произхождат от RP технологията (Requirements/resource planning - планиране на потребностите/ресурсите) е една от прилаганите логистични технологии. Основната цел на RP е съкращаване на количеството на запасите от материали, незавършено производство и готова продукция, съгласуване на графика на доставките с работата на отделните производствени звена и процеса на закупуване (Филипов, Ст. Г., 2019). С логистичните технологии и тяхното приложение се цели осигуряване на оптимални решения в логистичната система (Благоев, Бл., 2010). Логистичната технология се определя като стандартизирана последователност (алгоритъм) на изпълнение на отделни логистични функции, и/или процеси в логистичната система или в отделни нейни функционални области (Сергеева, В., 2004). Някои от тези алгоритми и поддържащите ги информационно управляващи системи са получили и нормативна регламентация. Такива са MRP I и MRP II, за които са разработени и утвърдени международни стандарти ISO (Стоянов, Ст. Хр., 2019). ER спомагат преодоляването на проблеми свързани с управление на складовите наличности, като:

• Забавяне на постъпването на материали;

• Натрупване на излишни складови запаси;

• Нарушаване баланса на доставката;

• Намаление на ефективността на производство;

• Сложност при процеса на контрол и анализ на производствената дейност;

Подобренията от въвеждането на ERP се изразяват в увеличаване броя на изпълнените поръчки, повишаване качеството на логистичното обслужване към клиентите, възможности за промени в обема на поръчките, съкращаване на времето от поръчката до доставката и други (Банабакова, В. К, 2019).

Фокусът на дисертацията е върху модула за продажби и дистрибуция. Цялата функция на този модул е да „продава стоки и услуги на клиентите на предприятието“. Основен обект на модула за продажби и дистрибуция е „материал“. Той може да се закупи, произведе, продаде, върне и/или прехвърли. Фигура 1.1. представя данните, които са част от този запис: продажби и логистика, количества за материали и доставки и други:



***Фигура 1.1****: Данни, част от записа за материал в модул за продажби и дистрибуция.*

Вторият най-критичен обект от гледна точка на продажбите и дистрибуцията е главният запис на клиента. Той представлява субект, на който се продават стоки и/или услуги. Част от атрибутите на запис на клиент са: името и адреса на клиента, условия на плащане, специфични опции за ценообразуване, които може да се прилагат само за този клиент, както и различните партньорски функции. ERP системите поддържат няколко версии на клиент. Първата партньорска функция представлява субектът, на който продаваме стоки и услуги (sold‑to party). Партньор за доставка (sold‑to partner) представлява мястото, където изпращаме стоки или услуги. С други думи, това е адреса за доставка на клиента, който може да е различен от адреса на купувача. Партньорът за фактуриране представя къде трябва да бъде изпратена фактура. Местоположението, на което се изпращат фактури, може отново да бъде напълно различно от това на адреса за доставка. Функцията партньор на платеца представлява субектът, който отговаря за плащането на фактура. В много случаи и четирите функции може да имат едни и същи данни, но в някои случаи тези четири функции са не само различни физически адреси, но и напълно различни обекти. Фигура 1.2. илюстрира на кратко четирите вида:



***Фигура 1.2****: Видове партньорски функции.*

*TODO: ценообразуване / поръчките / продажби / доставки*

### 1.1.2. Характеристика на системите за управление на транспорта

todo

## 1.2. Възможности за дигитализация на процесите по управление чрез прилагане на облачни технологии

В последните години облачните технологии се превърнаха във водеща тенденция в софтуерната индустрия. Те предоставят нов начин за изграждане на големи и сложни системи, като по този начин използват пълноценно съвременните практики за разработка на високо-качествен софтуер и налична инфраструктура. Това променя начина на проектиране, интегриране и внедряване на системите. Облачно базираните решения са проектирани да приемат бързо промените, да обслужват голям мащаб от хора и да бъдат устойчиви на всякакъв вид натоварване или хакерски атаки (Vettor, 2022).

Организацията Cloud Native Computing Foundation предлага следното определение: "*Технологиите, базирани на облак, дават възможност на организациите да създават и изпълняват приложения в модерни, динамични среди като публични, частни и хибридни облаци, чрез мрежи от услуги и микроуслуги. Качества на системите са устойчивост, висока наличност и достъпност, мащабируемост и управляемост, които са от критично значение за много от бизнес единиците. Автоматизацията на тези процеси позволява на инженерите да правят промени, с голямо въздействие, но с минимални усилия."*

Приложенията стават все по-сложни, като изискванията, от страна на потребителите, стават все повече и повече, главно насочени към бърза реакция и иновативни функции. Проблеми с производителността или повтарящи се грешки вече не са приемливи.

Предимствата на облачните системи поставят бизнеса една стъпка пред конкурентите. Бизнес системите се развиват от способностите на бизнеса да бъдат инструменти за стратегическа трансформация, която ускорява растежа на компанията. Облачно базираните системи се свързват главно с бързина (Smith, 2022).

…….

***Таблица 2.2****: Принципи за проектиране на облачни системи.*

|  |  |  |
| --- | --- | --- |
| име на български | име на английски | Описание (тук ще има доста цитати) |
| Разделяне на грижите | Separation of Concerns | всеки обект и модул трябва да бъде в своя собствена грижа и контекст |
| Капсулиране | Encapsulation |  |
| Инверсия на зависимостта | Dependency Inversion |  |
| Изрични компоненти | Explicit Components |  |
| Единична отговорност | Single Responsibility |  |
| Не се повтаряйте | Don’t Repeat Yourself |  |
| Устойчивост и невежество относно инфраструктурата | Presentation Ignorance |  |
| Ограничени контексти | Bounded Contexts |  |

## 1.3. Специфики при управление на поръчките от клиенти в производствено предприятие

Софтуерните приложения трябва да имат познание за потребителя или процеса, който ги извиква. Потребителят или процесът, взаимодействащ с приложение, е известен като принципал за сигурност, а процесът на удостоверяване и упълномощаване на тези принципали е известен като управление на самоличността (Vettor, 2022). Простите приложения могат да включват цялото им управление на самоличността в приложението, но този подход не се мащабира добре с много приложения и много видове принципали за сигурност. Windows, например, поддържа използването на Active Directory за предоставяне на централизирано удостоверяване и оторизация. Въпреки че това решение е ефективно в рамките на корпоративни мрежи, то не е предназначено за използване от потребители или приложения, които са извън домейна.

Удостоверяването е процес на определяне на самоличността на потребител. Упълномощаването е актът на предоставяне на удостоверено разрешение за извършване на действие или достъп до ресурс (Wike R, 2022).

Съвременните решения за самоличност обикновено използват токени за достъп, които се издават от защитена услуга/сървър (STS) на принципал за сигурност, след като тяхната самоличност бъде определена (Anil N, 2022).

Изискванията към STS:

• Защитават ресурсите;

• Удостоверяват потребителите;

• Осигуряват управление на сесии;

Важни функции на сървъра за самоличност са:

• услуга за удостоверяване, която да работи в централизиран процес;

• Единично влизане/излизане за множество приложения;

• Покрива индустриалните стандарти OpenID Connect и OAuth 2.0;

• Шлюз към Google, Facebook и др;

Удостоверяването чрез токени е механизъм без състояние, тъй като никаква информация за потребителя не се съхранява в паметта на сървъра или базaтa от данни, за разлика от бисквитките (Gichuhi , 2021).

Стандарт (RFC 7519) за уеб приложенията е JSON уеб токен (JWT). Той се състои от три части:

• Заглавна - JSON обект, кодиран във формат base64. Съдържа информация за типа на токена и алгоритъма за криптиране;

• Полезен товар - съдържа информация за текущия потребител (потребителско име, роля и др). Тук не трябва да се включват чувствителни данни, защотп лесно се могат да бъдат декодирани със публични сайтове като jwt.io;

• Подпис – Използва се от сървъра, за да провери дали токенът е валиден. Той се генерира чрез комбиниране на двете части (заглавна и полезен товар) заедно. Базира се на таен ключ, който само сървърът за удостоверяване знае. По този начин злонамерен потребител не може да фалшифицира валиден токен;

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1laWQiOiI0YTY2ZWNmNC1iZDdjLTQ3ODQtYmViOS1jZGM0MzQzZGY3MWYiLCJ1bmlxdWVfbmFtZSI6Im15QG15LmNvbSIsIm5iZiI6MTU5NjEzMzk3OCwiZXhwIjoxNTk2NzM4Nzc4LCJpYXQiOjE1OTYxMzM5Nzh9.W7k3UXA1g3TKxt-hR9a-mgCAcCsKjEyGTxBv5Dt79y8

***Фигура 1.3.1****: Пример за токен.*