# Глава 3. Изграждане и използване на облачна система за производствено предприятие Титан Цимент

За да се потвърди приложимостта на предложената архитектура и модел на софтуерна система, разработени във втора глава, системата трябва да бъде внедрена в реална работна среда. Едновременно с това е необходимо да се …

## 3.1. Обща характеристика на дейността на компанията

TITAN Cement е мултинационална компания със седалище в Атина, Гърция, и основен участник в глобалната индустрия за цимент и строителни материали. Корпорацията, която е основана през 1902 г., има операции в много страни, включително Европа, Близкия изток, Северна Америка и Африка. TITAN Cement Company S.A. произвежда, транспортира и разпространява широка гама строителни материали, като цимент, бетон, инертни материали и свързани стоки. Освен това те предоставят услуги на строителната индустрия.

Компания е специализирана в производството и доставката на бетонова смес, която се произвежда в централно съоръжение за дозиране. Терминът „готови смеси“ произлиза от факта, че тези смеси се произвеждат според спецификациите на клиента, което води до прецизен, висококачествен продукт, който може да се използва веднага след доставката.

За дозиране на бетон компанията използва големи централни съоръжения. Това дава възможност за прецизно и точно смесване, гарантирайки целостта на продукта. TITAN Cement произвежда смеси според изискванията за здравина, обработваемост и издръжливост. Често готовата смес се доставя в миксери и може да се използва веднага след пристигането. Това гарантира, че бетонът има най-високо качество, когато се излива, тъй като свойствата на бетона могат да се променят с времето.

Тъй като компаниите за готови смеси подготвят големи количества бетон на централно място, те могат да работят по-ефективно и в по-голям мащаб, отколкото ако бетонът се смесва на отделни строителни обекти.

TITAN Cement прилага строги мерки за контрол на качеството. Тъй като бетонът се комбинира в контролирана среда, TITAN Cement могат да гарантират, че съставът е прецизен и отговаря на всички приложими спецификации и стандарти. Централизираното смесване може да бъде по-благоприятно за околната среда от смесването на място, тъй като намалява отпадъците и позволява по-голям контрол върху използваните материали.

### 3.1.1. Основни бизнес процеси в компанията

Продуктите на компанията се използват за изграждане на къщи, инфраструктура, търговски и промишлени съоръжения, като по този начин отговарят на нуждите на нарастващото световно население за жилища, мобилност и икономическо развитие. Основната дейност включва производство и дистрибуция на цимент, инертни материали, готови бетонови смеси и асфалт.





Преход от хартиени документи към цифрови документи за доставки на бетон. Цифровите документи и цялата информация за доставка ще бъдат достъпни чрез приложението. Някои от плюсовете са:

▪ Без повече липсващи документи;

▪ Край на събирането и съхраняването на документи;

Те ще съдържат информация за поръчки, документи, фактури, резултати от тестове и др.

OnSite е нашето приложение, което ви помага да управлявате и проследявате напредъка на вашите конкретни доставки в движение в реално време.

Hub е нашият онлайн портал за управление, проследяване и свързване на цялата информация, свързана с продукта. Вашите поръчки, документи, фактури и протоколи от тестове – всичко това е на едно място.

Приветстваме ви в нашето безхартиено пътуване през нашите дигитални платформи.

### 3.1.2. Стимулиране на продажбите чрез цифрови технологии

Пакета от приложения стимулира прозрачността на данните, стандартизация в ERP, по-бързо и рентабилено планиране в заводите, създават иновативно решение, ориентирано към потребителите. Услугитe с добавена стойност, правят клиентите по-логистично интегрирани и по-добре оборудвани за да посрещнат предизвикателствата, свързани с устойчивостта. Целят да намалят материалните и логистични разходи, въглеродния отпечатък, да подобрят производителността и клиентския опит.

Проследяване на камиона с готов бетон по пътя му към строителна площадка, съхраняване на билети и протоколи едно място, с мобилно и уеб приложение. Благодарение на контрола, циментът се произвежда по по-устойчив и ефективен начин, който намалява отпечатък върху околната среда и осигурява безопасност на работниците. Обслужването на клиенти се рационализира чрез постоянната връзка с бек-офиса и превозвача.

Технологичния пакет се основава на авангардни технологии с отворен код, с най-новите програми и езици. Контейнерни услуги, работещи изцяло в облак и разпределени в множество регионални клъстери.

### 3.1.3. SWOT анализ

Фигура 3.x. изобразява SWOT анализ на готовността на Titan Cement за приемане на облачното решение, включващ силни страни, уязвимости, възможности и заплахи, като се цели отстраняване на слабостите и смекчаването на заплахите.



***Фиг 3.****: SWOT анализ*

Двигателите на бизнеса са интелигентни функции, устойчивост и системна интеграция. Бизнес резултатите се състоят от оперативна ефективност и интелигентни функции. Поради липсата на интеграция и автоматизация между системите, оперативната ефективност е отрицателно повлияна от забавената верига на доставки. Броят на интегрираните системи е KPI. Поради недостига на интелигентни функции в превозните средства за доставка, зашеметяващият ръст на приходите е свързан с интелигентните функции. Ключовият показател за ефективност е внедряването на мобилното IoT приложение.

## 3.2. Избор на технологични средства за разработка и операции

Това е решаващ избор, главно защото е почти необратим и разработчиците имат привързаност към предпочитаните от тях инструменти. В допълнение към изпълнението на целта трябва да се вземат предвид следните елементи:

• Проучване в Google Trends за данни относно популярността на определено технологично средство;

• Настоящи умения на разработчиците и тяхното по-нататъшно развитие;

• Oценка на разходите за използване на съществуващи и нови инструменти;

Таблици 3.3 представят анализ на сървърните технологии, подходящи за изпълнение на заданията.

***Таблица 3.3****: Сравнение на сървърни технологии за разработка.*



**.NET Core**

Статични или динамични: .NET Core, като рамка, поддържа както статични, така и динамични типове системи.

Инфраструктура: Най-подходящ за инфраструктура на Microsoft. Безпроблемна интеграция с Azure и други услуги на Microsoft. Поддържа разработка на различни платформи.

Общност: Има стабилна общност, особено в корпоративния свят. Поддръжката на Microsoft осигурява редовни актуализации и подобрения.

Производителност: Добро представяне. Известен с приложения на корпоративно ниво.

Крива на обучение: Умерено до високо, особено за разработчици, които не са запознати с C# и .NET екосистемата.

Набиране на персонал: умерено. Разработчиците с .NET умения са търсени, но те може да изискват по-високи заплати от някои други езици.

Облачен хостинг: Отличен, особено на Azure, където има безпроблемна интеграция.

**Node.js**

Статични или динамични: Node.js поддържа както статични, така и динамични системи.

Инфраструктура: Node.js е лек и може да се използва с почти всяка инфраструктура, включително Linux, Windows и macOS.

Общност: Много активна общност. Има множество пакети с отворен код, налични за Node.js.

Производителност: Известен с висока производителност, особено в IO-свързани приложения.

Крива на обучение: умерено. Разработчиците, запознати с JavaScript, ще го намерят по-лесно.

Набиране на персонал: лесно. Като се има предвид популярността на JavaScript, има голям набор от потенциални разработчици на Node.js.

Облачен хостинг: Отличен. Node.js е много мащабируем и се представя добре в облачна среда.

**PHP**

Статичен или динамичен: PHP се използва предимно за динамични уебсайтове, но може да се използва и за статични сайтове.

Инфраструктура: PHP работи на почти всички видове сървъри и е част от LAMP стека (Linux, Apache, MySQL, PHP).

Общност: Има масивна общност с обширни библиотеки и рамки, като Laravel.

Производителност: Подходяща за много приложения, но може да не е подходяща за приложения с много висока производителност.

Крива на учене: Като цяло лесен за научаване, особено за уеб разработка.

Набиране на персонал: лесно. PHP се използва широко от много години и има голям набор от PHP разработчици.

Облачен хостинг: Добре. Много хостинг доставчици предлагат PHP поддръжка. Въпреки това може да изисква повече конфигурация от някои други технологии.

**Java**

Статични или динамични: Java поддържа както статични, така и динамични системи. Въпреки това, той се използва по-често за динамични системи.

Инфраструктура: Java е независима от платформата, което означава, че може да работи на всяка машина, която има среда за изпълнение на Java.

Общност: Java има много голяма и активна общност. Освен това има обширни библиотеки и рамки като Spring.

Производителност: Висока. Java се използва в много критични за производителността приложения.

Крива на обучение: Умерено до високо. Самата Java е сравнително проста, но екосистемата може да бъде сложна.

Набиране на персонал: умерено. Java се използва широко от много години, но по-новите технологии може да са по-привлекателни за някои разработчици.

Облачен хостинг: Отличен. Java има добра поддръжка за внедрявания, базирани на облак, особено в приложения на корпоративно ниво.

**Python**

Статични или динамични: Python поддържа както статични, така и динамични системи. Въпреки това, той се използва по-често за динамични системи.

Инфраструктура: Python е гъвкав и може да се използва на практика във всяка инфраструктура. Често се използва в Linux среда.

Общност: Python има активна общност, особено в областта на науката за данни и машинното обучение.

Производителност: По-ниска от някои други езици, но често достатъчна за много приложения.

Крива на учене: Лесно. Python често се препоръчва като първи език за програмиране.

Набиране на персонал: лесно. Python е популярен и широко преподаван, така че има голям набор от разработчици на Python.

Облачен хостинг: Добре. Python се използва широко в облачни среди, но може да не работи толкова добре, колкото някои други езици за приложения с много високо натоварване.

Всички гореспоменати технологии имат своите силни и слаби страни и най-подходящата зависи от конкретния случай на употреба. Като се има предвид това, .NET Core е стабилна работна рамка с силна поддръжка, особено в среди на Microsoft и Azure облачен хостинг. Има голяма общност и осигурява висока производителност. .NET Core се вписва като най-подходящ вариант, въпреки по-стръмна крива на обучение спрямо други технологии.

Следваща стъпка е анализиране на Azure, Google Cloud и AWS по следните фактори:

* Съвместимост и интеграция с различни технологии;
* Поддръжка от общността и от трети страни;
* Мащабируемост;
* Цена;
* Крива на обучение;
* Набиране на персонал (Лесно намиране на квалифицирани специалисти);

**Microsoft Azure**

Съвместимост и интеграция: Отлична интеграция със софтуера на Microsoft и .NET Core. Той също така поддържа широк набор от други технологии.

Поддръжка от общността и от трети страни: Значителна поддръжка от трети страни и нарастваща общност, особено сред фирми, инвестирали сериозно в продукти на Microsoft.

Мащабируемост: надеждни възможности за автоматично мащабиране.

Цена: Ценообразуването е конкурентно, с редица различни модели на ценообразуване и налични опции. Предприятията със съществуващи договори на Microsoft могат да получат по-добри сделки.

Крива на обучение: Умерено, особено за тези, които вече са запознати с екосистемата на Microsoft.

Набиране на персонал: Умерено, тъй като пазарният дял на Azure е по-малък от този на AWS.

**Google Cloud Platform (GCP)**

Съвместимост и интеграция: Добра поддръжка за различни технологии. Интеграцията с услугите на Google е отлична.

Поддръжка от общността и от трети страни: Разрастваща се поддръжка от общността и от трети страни, но по-малка от AWS и Azure.

Мащабируемост: Известен с отлична мащабируемост, особено за големи данни, анализи и натоварвания с машинно обучение.

Цена: Често се счита за най-рентабилната опция за изчислителни екземпляри, но зависи от конкретния случай на употреба.

Крива на обучение: умерено. Някои уникални концепции, като проекти и инструмента за команден ред gcloud.

Набиране на персонал: По-трудно, тъй като GCP има по-малък дял от облачния пазар в сравнение с AWS и Azure.

**Amazon Web Services (AWS)**

Съвместимост и интеграция: Поддържа широка гама от технологии. AWS разполага с широк набор от услуги, които се интегрират добре помежду си.

Поддръжка от общността и от трети страни: Има най-голямата общност и най-много инструменти от трети страни поради статута си на лидер на пазара.

Мащабируемост: Силно мащабируеми, с множество услуги, специално проектирани за мащабиране.

Цена: Ценообразуването може да бъде сложно, но AWS предлага широка гама от опции и е конкурентен в повечето области.

Крива на обучение: Умерена до висока, в зависимост от използваните услуги. AWS има много услуги и функции.

Набиране на персонал: Най-лесният сред трите поради доминирането на пазара на AWS.

Azure предлага отлична интеграция с .NET Core и може да използва всяка съществуваща връзка, която организация има с Microsoft като активна директория, мейл сървър и други. AWS и GCP също поддържат .NET Core, като всички три платформи имат силни и слаби страни. Най-подходящият избор, зависещ от конкретен случай на употреба и инфраструктура е Azure.

## 3.2. Организационни аспекти при реализация и експлоатация на системата

### 3.2.1. План за реализация и внедряване на системата

Един от основните моменти при разработката на софтуерната система е планът за реализация и внедряване. Както е известно, функционалността на софтуерните системи е възможно динамично да се променя във времето, при което се налага извършването на допълнителни дейности. Поради това е необходимо, както планиране на реализациите и версиите, така и характеристики, които се реализират в тях, заедно с разпределение във времето. На база изложеното до момента, е разработен съкратен вариант на работния продукт на база сценарии за взаимодействие. При необходимост той може да се разширява и допълва според потребностите и стратегическите виждания на ръководството.

Планът за реализация е представен в табл. 3.1. Заложените срокове са ориентировъчни и са подходящи за малки до средни по размери организации.

Таблица 3.1.  
 План за реализации на проекта на софтуерната система за управление производството на софтуер в технологични стартиращи компании.  
(разработка на автора)

| **Етап / Реализация** | **Идея на етапа** | **Включени сценарии / характеристики** | **Примерен срок** |
| --- | --- | --- | --- |
| **Детайлно проектиране** | Детайлизация на проект, оглеждане на 360°, конвенции. | Етап за детайлно проектиране с детайлизация на проекта на база текущия проект, концепцията и избран подход | Една седмица |
| **Скелет на проекта** | Скелет на всички компоненти и минимум работещ сайт достъп, меню и тестова редакция на продукти. | База данни с примерни данни и без оптимизация; Софтуерен проект и физически компоненти само със скелет на класовете; Настройки на сървъри, добавени базови компоненти. Структура на интерфейс; Логване на потребител; преглед/дефиниция на продукт; | Две седмици |
| **Компоненти за логика и данни** | Разработка на компоненти за данни и бизнес логика | Преглед/дефиниция на продукти; Достъп до данни; Усъвършенствани с новите модули и проиграване на модулите; | Две седмици |
| **Дефиниране на проект и план** | Всички функции за дефиниране на проект, съдържание, процес и план; изпитване компонентите; | Сайт, в който потребител може да дефинира продукт, проект, характеристики, и т.н., подреждане и оценка на характеристики, избор на процес и скали за, дефиниция реализации, разпределение на характеристики, изготвяне на план; | Три седмици |
| **Работа по задачи и тестване** | Планиране и изпълнение на итерации и задачи, тестване, проблеми, дефекти | Дефиниране на итерации, задачи, списък рискове и задачи по тях, смяна на статус, минимален план и отчет на време по тях; дефинира проблеми, дефекти, конвертиране, обработка на дефекти и рискове; | Три седмици |
| **Детайлен отчет в работата** | Детайлизация на план и отчет по работата и итерациите | Записва изразходвано и оставащо време работа по задачи. Лично табло за статус. Отчитане разход на време и средства. Записки и документация към задачи, итерации и всички обекти. Записва изводи от отразяващото подобрение в итерация; | Две седмици |
| **Приключване на реализация** | Да може да се приключи реализация и отчетност. Да се оформи отразяващо подобрение. | Справки за статус и очаквано завършване. Корекция на оценки за итерации/реализации, дефиниция на реализирани продукти. Записва изводи от отразяващото подобрение в реализация. Назначава задачи от тях. Добавяне на заявки от клиент, трансфер в характеристики. | Две седмици |
| **Пълен работен цикъл и администрация** | Изчистване за пълен цикъл разработка. Администриране на клиенти и служители. | Подобрение за тестване и записки, експорт на списък характеристики за лесна документация. Тагове към всички обекти. Подобрени екрани за отчетност на време и средства. Клиенти, представители, администрация на потребители, настройки, продцеси, скали и т.н. Детайлни данни за служители – история, ставки, умения, календар за време и др. | Три седмици |
| **Договаряне** | Да може да се изготви оферта за клиент и дефинира договор; | Добавяне на нов договор или оферта, определяне усилията за тях, съобразяване с капацитет. Изготвяне на Оферта, отпечатване, дефинира договор/план с параметри; детайлно планиране на времето по реализации/итер и съответствие с капацитет. | Две седмици |
| **Отчетност** |  | Компонент за отчети, стандартни отчети за ефективност, приключване на проект. Подготвя информационни табла и други отчети. | Две седмици |
| **Поддръжка** | Дефиниране на поддръжка, мониторинг и работа по поддръжка | Дефиниране на поддръжка – координатор/главен проектант дефинира изисквания, условия, мониторинг при поддръжката; Следене и попълване на заявки за поддръжка. Конвертиране на заявки в дефекти, задачи или характеристики за обновяване. Опис на внедрени версии. | Две седмици |
| **Финални** | Други екстри; приключване | Потребител описва мнение за удовлетвореност от внедрена версия и оказана поддръжка. Отчети за дефекти, дял поправени, за поддръжка | Две седмици |

Планът е за около 26 седмици (или 6 месеца) и подлежи на детайлизация. Целта е в първите три месеца да има годна система за оперативна работа по проекти. Когато след три месеца е готова, екипа може с реални данни да изпита системата.

По отношение на процеса по **внедряване** са набелязани комплекс от основни стъпки и мерки, които трябва да се извършат при внедряване на софтуерната система в дадена технологична стартираща компания:

– Проучване и гарантиране, че технологичната стартираща компания и екипът й е готов за прилагане на подхода;

– Разработване на индивидуален подход, процес, техники, работни продукти в компанията;

– Бързо тестово проиграване на горните, за да е сигурно, че работят;

– Инсталиране на хардуер – сървъри, клиентски станции, мрежа, защита и т.н.

– Инсталиране на софтуер – системен и базов софтуер, приложения на софтуерната система;

– Настройка на потребители, права, шаблони за процес, скали, нови опции и т.н.

– Въвеждане /трансфер/ на данни от текущи и тестови проекти;

– Интеграция със съществуващ софтуер (ако е нужна)

– Проиграване и тестване с екипа в реални условия; финални корекции и настройки;

– Одобрение на внедрената софтуерна система от ръководството;

– Въвеждане на всички данни за текущи проекти и за желани исторически данни;

– Защита на средата и софтуерната система, създаване на план за архивиране и за справяне с рискове.

След внедряването, всяка система изисква съответни мерки преди и по време на нейната **експлоатация**. Набелязани са комплекс от мерки по използване и поддържане на софтуерната система. Тъй като предлаганият от нас подход за разработка на софтуер отделя специално внимание на работата в екипа и с софтуерната система работят участниците в екипа, то следва да се включат мерки по подготовка на екипа и използване на софтуерната система от него. Тези мерки следва да се включат и като част от конвенциите (правилата) за работа по проекти. За първите приемаме, че могат да включват следните обичайни мерки в отделните етапи от отделните роли:

– **Администриране и координация**: администратор с помощта на главен проектант и координатор настройват софтуерната система, шаблони на процеси, скали и други опции; администратор и координатор осигуряват на всеки потребител адекватен достъп и данни за служителите, включително квалификация, история, работно време, ставки и т.н.; координатор и възложител се грижат за адекватно дефинирани продукти/проекти.

– В **началото** на всеки софтуерен проект се оглежда заданието според случая и планира проекта: екип/координатор описва условия, характеристики и параметрите им; екипът използва техники за оценяването му; координатор, възложител и главен архитект планират реализациите; изготвяне на оферта, дефиниране на договор и се планира детайлно първа реализация.

– **Ежедневно** екипът в технологичната стартираща компания има следните задължения: след 15 мин. среща се отразява всичко от срещата за статуса на задачи/проект, проблеми и бележки, лично или от координатор; преглеждат се всички новопоявили се новини, проблеми, рискове и т.н., обработват се и се назначава отговорен за тях; в началото на работа се отбелязва започването и темата на работа; в края на задача и ден се записва изработеното време и очакваното време до завършване на задачата или задачите, по които се работи.

– Поне веднъж или два пъти **седмично**: преглежда се статуса на всеки проект, всички рискове, проблеми, новини; преглеждат се отчетите за очаквано завършване и прогнози; при проблем (сигнализация в жълто или червено, или друго) се обсъждат мерките, които трябва да се вземат и се поставят лични задачи за справяне; при значим проблем с извънредна работа, забавен ход, или необходима промяна на договор, част от работата се спира до предоговаряне; преглеждат се всички висящи новини, проблеми, рискове, дефекти и други записки, обработват се и се назначава отговорен за тях; базата от данни се архивира и съхранява на отделен сървър или носител; да не се допуска да има задачи, които не са отчетени и оценени.

– В **края на всяка итерация и реализация**: дефинира се продукта за реализация; записват се бележките от обсъждането с клиента; отразяват се в плана; допълва се отчетност по итерация, задачи, записки, характеристики и др.; извеждат се отчети за представяне в завършената итерация/реализация; отразяват се резултатите на отразяващото подобрение; дават се задачи; БД се архивира на технически носител и се съхранява извън офиса.

– В **края на всеки проект**: допълва се незавършената отчетност по проекта; описват се реализирани продукти, версия; финансови отчети; отразяват се резултатите на отразяващото подобрение; дават се задачи; ако са необходими корекции в процес и правила, предефинират се; проектът се приключва и архивира.

– При **аварийни ситуации**, когато софтуерната система не може да работи: екипът продължава работа по същество на проекта; ползват се работни продукти ръчно; администраторът коригира софтуерната система веднага; в най-близко време данните от работните продукти се пренасят в софтуерната система.

По отношение на **оперативния мениджмънт на екипа**, работещ с софтуерната система, в първа глава е изяснена важността на човешките ресурси и управлението им в технологичната стартираща компания, и бяха представени необходимите качества за работа с гъвкави подходи[[1]](#footnote-2). Идеологията и духа на тези подходи изискват голяма тежест да се отдава на работата в екипа. Както отбелязват някои автори (Fowler, 2019) гъвкавите подходи не са подходящи за всички ситуации и изискват определени предпоставки и мерки. Необходимо е да се изяснят някои съображения и изисквания към екипа на технологичната стартираща компания и външните участници, и това трябва предварително да се има предвид.

**Първо**, участниците трябва да са готови/подходящи за прилагане на гъвкав подход.

**Второ**, участниците трябва да познават процеса и използваната методология за разработка. Според нас, една подходяща техника за подготовка е т. нар. „Тренинг в мини процес”. В нея се проиграва процесът и методологията с най-прост примерен продукт, евентуално два пъти по 1-2 часа, за да са наясно новите членове на екипа с методологията. Още по-добър резултат може да се получи, ако те могат да проиграят, заедно с това, и работата с софтуерната система.

**Трето**, софтуерната система трябва да се познава и да може да се ползва. Въпреки, че целият екип може да взима участие в системата, повечето работа относно процеса и по-сложните функции касаят роли координатор и главен архитект. Затова тяхното обучение и трениране е ключово. То е и ключово, защото те се грижат за екипа и го насочват. Програмистите се очаква да работят „рамо до рамо”, да имат близка комуникация и да ползват съвети от останалите, така че те могат бързо да бъдат обучени на място или да се само обучават един от друг. Системата е проектирана така, че да дава възможност за бързо получаване на кратка помощ и инструкции за специфичните опции.

**Четвърто**, потребителите трябва да имат минимум технически умения за работа с клиентско уеб приложение. Може да се смята, че в днешни условия това не е проблем за вътрешния екип, и малко вероятно за външни възложители/потребители. Необходимо е само кратко въведение в начина на работа с приложението. Ако възникне проблем или потребители не са допуснати до софтуерната система, то координаторът с подходяща комуникация с тях може да изпълни тези задачи. Администраторът и вероятно координаторът е нужно да владее администрацията на продукта и да познава администриране на СУБД MS SQL Server или MySQL, и уеб сървър IIS, Apache или nginx, което включва регулярно архивиране на БД (Куюмджиев, 2019), настройки за достъп и оптимизация.

**Пето**, важни са уменията да се дефинира и пише точно, ясно и кратко, и да се избягва прекалена формалност. Софтуерната система и подходът са замислени като опростено средство и е излишна загуба на време извършването на повече административна работа. И не на последно място, справяне със стреса и намаляване на причините за него. Например, наличие на прекалено натоварване в задачи, видимост на прекалено много несвършена работа и др. Някои прости мерки в това отношение са полезни – интерфейсът показва прогрес, личното табло го показва и скрива дългите списъци над определен размер, фокус в работата, избягване на многозадачност и добавянето на нови характеристики поне в започната итерация, както и ритуали подкрепящи прогреса.

По отношение на **администрирането на системата** може да се приеме, че то трябва да се извършва само от ролите администратор и главен проектант. Важен компонент в софтуерната система е административна част в клиентското приложение и менюта „Администрация” и „Профил”, като чрез първото може да извършва:

– администрация на потребители – добавяне, активация, промяна на данни и парола;

– администрация на служители – пълни данни съгласно модела за служител и историята му – данни за контакт, квалификация, възнаграждение, параметри за работата;

– назначаване на потребител към проект с определена роля;

– дефиниране на шаблон за процес;

– дефиниране на рангова скала;

– дефиниране на списъци или добавяне на опции в разширяеми списъци;

– поправка на често срещани грешки – местене на задачи между итерации, характеристики между реализации, проблеми между проекти, корекция лог и подобни дефекти;

– логическо архивиране – изключване от работа на елементи маркирани като архивирани и улеснено цялостно архивиране на проект/продукт.

Чрез меню „Профил” потребителят може сам да сменя данните си и паролата си.

В хода на експлоатация на софтуерната система могат да възникнат проблеми от различни естество – хардуер, софтуерно осигуряване и др., и да прекратят ползването й. **Възможните рискове и адекватни противодействия** са следните:

– срив в БД – регулярно архивиране на БД и процедура по възстановяване;

– срив в сървър – подмяна или местене на приложенията на нов; смяна на настройки; виртуализация;

– пробив в сигурността – ползване на сигурни протоколи – SSL/TLS, VPN, забранено кеширане, сигурност на работни станции и уеб сървър, ползване на силни пароли, и т.н.

– друг срив на системата – работата по проектите може да продължи с шаблони от работни продукти във файлове, а по-късно да бъдат въведени в софтуерната система;

За противодействие на рисковете е необходимо регулярно да се извършва преглед на следните мерки: изработване на план за справяне с рискове относно софтуерната система; дефиниране на специален проект за технологичната стартираща компания и процеса; регулярно документиране в специалния проект на проблеми, дефекти, рискове и отговорно поставяне на задачи за разрешаването им; регулярно отразяващо подобрение в планове и конвенции за рискове.

Включването на управление на риска като реален проект и отговорното му възприемане под формата на задачи с пряка отговорност е предпоставка за по-надеждна работа.

Като потребители на софтуерната система могат да са технологични стартиращи компании с размер на екип от 2 до 9-12 човека. При по-голям персонал, екипът за един проект трябва да бъде в лимита и да няма пряка връзка и зависимости между проектите, в случай че член на екипа участва в няколко проекта едновременно. Едно възможно решение за по-голям екип, работещ по един софтуерен продукт, е разделение в по-малки екипи по отделни проекти със синхронизирани изисквания. Примерно, подобни проекти може да са относно:

– общи компоненти на продукта, валидни за всички версии;

– разработка на отделни версии на продукта;

– отделен екип да прави съпровождане на продукт;

– средства за продажби, автоматизация и обслужване на клиенти;

– индивидуални внедрявания на база основния продукт.

Синхронизацията между проектите може да се осъществява по следните няколко начина. Първият е, чрез синхронизиран дизайн на интерфейсите (Poppendieck, 2014) като инструмент. Втора идея е от подхода SCRUM да се правят регулярни ежедневни срещи между главните проектанти на отделни екипи. Аналогична идея съществува и в подхода LSD, където може да се приложи инструмент за вземане на подобни решения базиран на множество опции. Също така, възможна е ротация на членовете на екипи в различни области с цел пренасяне на опит, развитие и натрупване на разнообразен опит, разнообразяване в работата и други.

Софтуерната система се прилага за различни проекти – на външен възложител или собствени, с фиксирани или гъвкави параметри. Според нас, е възможно приложението в разработка на интелектуални продукти различни от софтуер, но само ако избраният подход е удачен за тях. По-съществените ползи са:

– видимост на продукта и проекта в една система;

– лесна приспособимост в разработките и други ползи от инкрементален адаптивен подход;

– усъвършенстване на бизнес процесите;

– повишаване на качеството на управление чрез усъвършенстван процес;

– измерване на резултатите от дейността с опростени отчети;

– увеличаване на производителността и по-добра ефективност на специалистите;

– опростена система на работа;

– автоматизация за повечето работни продукти;

– формално вградени механизми за подобрение, добро качество и удовлетворение на клиентите;

– подходящо структуриране на персонала и разтоварване от административна работа;

– средства за разпределение на работата по тежест във времето;

– средства за прогнозиране спазването на сроковете.

Когато основната дейност на технологичната стартираща компания е производство и поддръжка на софтуер с конкурентно качество, цена, време за доставка, както и съпровождане, то управлението на компанията трябва да организира адекватно софтуерно производство. За целта е необходимо да се направи избор на подход и процес, като се опишат залегнали техники и практики, както и въпроси на управлението и работа в екип. Важен момент тук е изборът на подходящи подходи за организация на софтуерния процес, адекватни за обекта на управление, на чиято база да се осъществява управлението и чрез софтуерната система.

### 3.2.2. Особености при експлоатация

Разработването на система за управление разработката на софтуер е продължителен и сложен процес, който се нуждае от планиране и спазване на определени принципи. Прилагането на софтуерната система налага както технологични, така и организационни промени, които са породени и от необходимостта от прилагане на нова схема на управление, изискваща по-тясно сътрудничество между компанията и клиента. Необходимо е при изграждането на софтуерната система да се спазва и принцип за преизползваемост, т.е. повторната употреба на готови компоненти, като още в началото се предвидят възможности за повторно използване на компонентите чрез подходящо проектиране.

В разглежданата компания БитПайъниърс разработването на софтуерната система задоволява следните бизнес потребности:

– Автоматизиране на тези дейности по разработване на софтуер, които не са осигурени с подходящи инструменти;

– Обединяване на съществуващите системи в обща система за управление на разработка на софтуер;

– При тестването предлагаме да се използват автоматизирани средства за управление на тестването. Голяма част от потребителските изисквания се описват в текстови документи и впоследствие ръчно се преобразуват в тестови сценарии, при което използването на автоматизирани средства за управление на изискванията повишава производителността. Едновременно с това се повишават възможностите за бърза адаптация спрямо често променящи се и непредсказуеми изисквания от страна на клиента. Част от процесите по планиране на даден софтуерен продукт може да продължат да се извършват ръчно, съобразно конкретната обстановка.

– Прилаганите подходи в БитПайъниърс дават възможност за лесна бъдеща интеграции с нови софтуерни инструменти или за получаване/изпращане на данни, постъпващи от други бизнес партньори при съвместна работа по проекти (например при аутсорсинг).

Разработването на софтуерната система решава посочените проблеми като осигурява относителна независимост между използвани платформи, софтуерни средства и програмни езици при разработването на дадено софтуерно решение. За целта се създават нови компоненти, които да предлагат необходимата функционалност.

При използването на система за управление разработката на софтуер в технологична стартираща компания е необходимо да се избере и съответен подход за разработка, подходящ за конкретната ситуация. Според нас, ако се използва подходяща софтуерна система, но се прилага неподходящ подход за управление на процесите на разработка, то няма да се постигне висока ефективност на работа и дори да се стигне до ситуация на неефективна работа по проекта. Поради тази причина приемаме, че изборът на подход при разработката на софтуер е също толкова важен, колкото е и използваната софтуерна система, прилагана за автоматизиране на отделните дейности по разработката на софтуер.

В практиката широко са разпространени няколко подхода, всеки със своите предимства и недостатъци при прилагане в различни ситуации. Затова обоснованият избор на подход може да се окаже от голямо значение за реализацията на софтуерен продукт от технологичната стартираща компания или за нейния провал.

В случаи на малка компания, като БитПайъниърс, с голям дял индивидуални („бутикови”) разработки, според нас, е подходящо преобладаващо да се използват т.нар. „гъвкави методологии” (виж Таблица 3.2).

В повечето случаи възможностите на БитПайъниърс да поема проекти с извънредно висока променливост на изисквания, непрестанно участие на потребителя на място и висок формализъм, са ограничени. В тези случаи, според нас, е подходящо прилагането на опростен подход за малък екип с ниска до средна формализация (по-висока за собствен продукт). Времево, проектите са за 1-12 месеца. По своя срок на изпълнение и размер проектите условно можем да ги разделим в следните категории:

– микро поръчки, със срок на изпълнение в рамките на 1-2 седмици с много малко изисквания;

– малки или прости проекти, със срок на изпълнение до 3 месеца (външен проект, малко компоненти);

– големи, по-сложни проекти с повече модули и документация, със срок на изпълнение от 3 до 12 месеца. Още по-големи проекти може да се се разделят на няколко големи проекта. Тук влизат почти всички проекти за продукти/компоненти.

В случаи на **микро поръчки**, предвид ограниченията, според нас, подходящи са методологиите SCRUM, CC, LSD и DSDM. Според нас, методологията UP в тези ситуации не е подходяща, тъй като SCRUM се явява опростен вариант на UP. Същото се отнася и за методологията XP, поради не толкова голямата променливост на изисквания. За малки проекти трябва опростена адаптация на CC или SCRUM с опростени механизми. Според нас, най-удачен подход е CC, комбиниран с някои практики на SCRUM/LSD/XP като се разчита на кратки реализации и итерации (или сливането им), със срокове на проекти под 3 месеца. По този начин по-силен акцент може да има над техники за работа в екипа и гъвкавост.

За **малки или прости проекти** предвиждаме да се използват CC, SCRUM, LSD или DSDM, но с повече реализации/итерации, формализация и повече работни продукти, тъй като след като продуктът е по-голям, то и изисква последваща поддръжка и преизползване за компоненти. В този случай трябва да има отличителна разлика между реализации и итерации – реализация приблизително всеки месец, с 2-3 итерации в нея.

За **големи проекти** – продукт или компонент, предлагаме да се приложи CC, SCRUM или LSD. Целта е да има иновативен и по-формализиран процес, с повече моделиране и документация, тъй като произведеният продукт трябва да се подновява и поддържа. За продукт предлагаме да има наличен един минимален пакет модели и документация, както и по-широк набор задължителни практики. Според нас е подходящо реализациите да са на един или два месеца с няколко итерации в тях. За още по-голям продукт е необходим съответно и по-формализиран процес, но без загуба на иновативност. За него може да се ползва SCRUM, LSD или CC с елементи от UP/DSDM.

**Услугите** **по поддръжка** предвиждаме да се реализират като малък проект за клиент с лек подход, минимум реализации и изисквания. Поддръжката има друг формат на работа, спрямо разработката, и се реализира в проект с лек стандартизиран подход и документация. Итерациите в поддръжката са едностепенни и може, обичайно, да са едномесечни. Ако е необходимо обновяване на продукт, изискванията се добавят към дадена итерация или се прави нов проект. За разлика от предходните, тук може да се добавят нови изисквания по време на итерацията, обичайно като промени или дефекти, като вторите се обработват възможно най-скоро в самата итерация. Поддръжката е натоварена с обновяване на модели, продукт, документация за поддръжка и за потребителя. В табл. 3.2 са показани избраните подходи.

Таблица 3.2.   
Адекватни подходи за отделните дейности на компанията БитПайъниърс  
(разработка на автора)

| **Вид дейност** | **Подход за малки проекти** | **Процес, фази, итерации за малки проекти** | **Подход за големи проекти** | **Фази и итерации на жизнен цикъл** |
| --- | --- | --- | --- | --- |
| **Клиентски проект** | CC, SCRUM с кратки цикли | Лек; Начална, Проектиране, 1-2 седмични итерации, Внедряване | SCRUM, CC, LSD, DSDM | Начална, Проектиране, 4-6 седмични реализации, Внедряване |
| **Собствени продукти** | Малко вероятно, SCRUM, CC,LSD | Планиране, Проектиране, 2-4 седмични реализации, Внедряване | SCRUM/UP, CC, LSD, DSDM | Планиране, Проектиране, 4-6 седмични реализации, Внедряване, Поддържане |
| **Компоненти** | Същите | Същите с модификация | SCRUM, CC, LSD | Същите с модификация |
| **Поддръжка** | Модификация с регилярни и възник.задачи | Месечна поддръжка с итерации и реализации по необходимост | Модификация с регулярни и възник. задачи | Месечна поддръжка с итерации и реализации по необходимост |

Възможно е да се избере комбинация от два подхода. Например единият може да се прилага при малки клиентски проекти, а вторият може да бъде изцяло за вътрешни продукти (и компоненти), както и за най-големите клиентски проекти. Причината е, че често успешни големи проекти (по-рядко малки) продължават в някои от следните възможности[[2]](#footnote-3):

– поддръжка поне една или две години след завършване;

– заявки за обновяване в следващите една до две години;

– основа за разработване на нови компоненти – например за чести проблеми;

– продукт или вертикална система за преизползване, който компанията БитПайъниърс може да изгради като собствен на база опита си в проекти;

– база за преизползване на готови решения в дадена област и проблем;

За малък проект реалистично се очакват само следващи микро заявки или малки проекти за обновяване, или дори компоненти на тяхна основа (ако технологичната стартираща компания реши), но рядко поддръжка. Поради това, според нас, е подходящо подходите да се разделят в две основни категории – една за малки проекти и една за продукти и големи проекти. За всеки конкретен проект БитПайъниърс трябва да определи границата между двете, както и случаите, при които е подходящо да се приложи определен подход. Според нас, за малки проекти най-подходяща е методологията CC. За големи проекти и продукти препоръчваме да се избере CC с повече техники и практики, както и елементи заимствани от методологиите SCRUM, UP, DSDM и LSD. В този случай, акцентът може да е върху характеристиките иновативност и качество, гъвкавост и „компромиси”, за се вмести БитПайъниърс в пазарни и вътрешни ограничения, както и да създаде максимално „очароващ продукт“. От организационна гледна точка с висока степен на важност са въпросите, свързани с **процеси, техники, практики и работа в екип**.

В заключение, може да се обобщи, че в зависимост от конкретната ситуация са подходящи различни подходи при разработката на софтуер, с цел работата по тези проекти да е организирана по оптимален начин в компанията БитПайъниърс. Заключенията, в резултат на изследванията, могат да се използват и от други сходни по размер и дейност компании. В следващите точки от изследването, спираме своето внимание върху необходимия инструментариум за физическа реализация и управление на разработването на системата и възможностите за използване на готови виртуални инструментални средства.

## 3.3. Физическа реализация на системата

### 3.3.1. Избор на технологични средства за реализация на системата

За физическата реализация на софтуерната система за управление на производството на софтуер в технологични стартиращи компании е необходимо да се подберат конкретни технологични средства за разработка и софтуерни компоненти. Предвиждаме софтуерната система да се интегрира с програмното осигуряване, използвано в обекта на приложение и по този начин да се реализира логическия модел, предложен във втора глава.

От голямо значение за изграждане на софтуерната система са следните етапи, които включват определяне на инструменталните средства и приложното програмно осигуряване, които ще се използват:

1. Изграждане на слоя с данните:

– избор на релационна СУБД, която ще се използва, придобиване и инсталиране;

– изграждане на структурата на БД.

2. Изграждане на слоя за бизнес логиката:

– избор на програмен език, среда за програмиране и софтуерен фреймуърк за ускоряване разработката на програмните модули;

– разработване на програмните модули, които осъществяват връзката между слоя с данните и потребителския интерфейс;

– прилагане на тестове за осигуряване на качеството на софтуера.

3. Изграждане на представителния слой:

– избор на облачна услуга, хостинг услуга или уеб сървър, инсталиране и конфигуриране;

– избор на софтуерен фреймуърк за разработване на динамични уеб страници;

– разработване на уеб интерфейса на системата и на базовите интерфейсни компоненти.

Изборът на системно програмно осигуряване и по-специално изборът на операционна система е важен момент при разработката на софтуерни системи, но на този етап този въпрос остава на заден план. От една страна, наличен е софтуер за виртуални машини, който дава възможност под управлението на една ОС да работят множество виртуални компютри, всеки със своя собствена ОС, а от друга страна включването на Windows Subsystem for Linux (WSL) като един междинен слой, дава възможност за стартиране на Linux изпълними файлове под Windows 10 и Windows 11, които са най-разпространени сред десктоп системите. От тази гледна точка, следните базови въпроси и проблеми са извън обхвата на труда, тъй като в изложението се концентрираме само върху софтуерната част от една информационна система и освен това в самата компания има налична техника, която може да се използва споделено. Това са въпросите по: избор и закупуване на компютърна техника - сървъри, мрежови устройства и инсталиране; осъществяване на връзката с Интернет - скорост, избор на доставчик; системно програмно осигуряване - избор и закупуване на операционна система за сървъри и клиенти, инсталиране, конфигуриране и поддържане.

За да се реализира практически софтуерната система е необходимо следния софтуер (програмно осигуряване) и инструментариум от средства:

1. Системен софтуер:

– сървърна операционна система: MS Windows Server 2019/2022 (налична във фирмата);

– клиентска операционна система: Microsoft Windows 10/11 (налична във фирмата), или друга, на която може да се ползва браузър със спецификацията, дадена по-долу;

2. Базов софтуер:

– система за управление на БД: Microsoft SQL Server Standard 2017/2019;

– уеб сървър на приложен сървър: Microsoft IIS 10;

3. Потребителски софтуер:

– интерфейс: браузър (Google Chrome, MS Edge, Mozilla FireFox, Opera, или друг с поддръжка на HTML5, CSS 3.0, JavaScript/ECMAScript 2021;

– програмна среда: ASP.Net 4.8/ASP.Net Core 6.0, C# 10.0 и VB.Net 16.9+ – за уеб приложение;

– програмни компоненти:

• ASP.Net AJAX – за поддръжка на AJAX интерфейс;

• FusionCharts XT за графики (http://www.fusioncharts.com/);

• вътрешни компоненти за библиотеки относно отделни слоеве.

– интерфейс за програмен достъп – уеб услуги за достъп (SOAP API).

В заключение, предложения софтуер може да е предпоставка за реализиране на софтуерната система, но не е напълно достатъчен. Необходимо е използването на допълнителен инструментариум за управление на разработването на системата.

### 3.3.2. Инструментариум за управление на разработването на системата

Развитието на ИТ индустрията, повишаването сложността на софтуера, появата на нови методологии и процеси за разработване води до налагането на софтуерни инструменти за управление на разработването на софтуерни продукти. В най-общ план тези средства може да се класифицират по следния начин:

1. Утвърдени универсални инструменти за управление на проекти (MS Project 2019/Online, Scitor Project Scheduler 8.5, Oracle Primavera P6, Project Libre и др.) с богат набор характеристики и настройка с добавки, които покриват всякакви проекти от всякакви области. Обичайно подхода им е на база PMI[[3]](#footnote-4) с добре дефиниран проект.

2. Специализирани за управление на проекти по даден подход (MS Visual Studio Team Services (VSTS)/Azure DevOps Services, Jenkins, Jira и др.). Те имат подобрени средства и техника само за него, понякога простота на работа и специфичен интерфейс, уеб достъп, възможност за преработка на код и БД. Някои са само добавка към горните.

3. Специализирани за софтуерни проекти/продукти по определена методология, обвързани с една или група методологии. Характерни черти са тясната специализация със софтуерна методология, покриват специфични функции за софтуерното производство, съвместна работа на екипа в системата с различни роли и права за достъп. Някои от тях са уеб базирани, дават възможност за индивидуална настройка или модификация, или интеграция с утвърдени продукти за управление на проекти (например MS Project).

Съществува голямо разнообразие на продукти във всяка една категория и ние считаме за подходящо, в случай че има съответни финансови средства, използването на MS Project 2019/Online. При липса на достатъчно средства считаме за подходящо използването на специализирани средства от нисък клас, които притежават общи полезни средства – InstantTeam (instant-team.com), Helix ALM (perforce.com), Project Tracker (projectmanager.com), и др. Всички тези продукти имат различен начин на действие, но общата функционалност – подход, групова работа, мобилност, уеб достъп, опции за индивидуализация и модификация, интеграция с популярни продукти са налице, т.е. те имат множество функции за проект, задачи, групова работа, дефекти и т.н.

Според нас, използването на MS Azure DevOps Services има следните предимства, които го правят подходящ за използване при разработката на конкретната система:

- управление на ниво продукт с отделни, добре дефинирани реализации;

- удобен специфичен визуален интерфейс с влачене, менюта, страници, подредба;

- приоритет на функционалности на база % бизнес тежест;

- тематично групиране на характеристики и филтриране/отчитане на база темите;

- следене на задачи в итерация, отчитане на отработено време, проблеми, бъгове;

- специфични лични списъци задачи, календар работно време;

- йерархична организация на характеристики за програма и влагането им в множество продукти, поставяне на тежест на групи; Отчети за постигната бизнес стойност;

- уеб базирана система за отчети и споделянето им; настройваем генератор отчети;

- тематични отчети за скорост, променливост изисквания, постигната бизнес ценност, възвращаемост като съотношение на % бизнес тежест към % вложени усилия;

- графика за очаквано завършване и прогнозиране с линеен тренд, тренд от оптимистичен и песимистичен опит; предсказване на очакван край с емпирични данни за прогрес;

- интеграция с продукти за следене на дефекти като BugZilla и JIRA; експорт в Excel, пълен програмен достъп през SOAP API, e-mail уведомления.

Относно използването на софтуерни инструменти за специфични функции на производството и поддръжката на софтуер, при разработката на системата, могат да се използват инструменти в следните направления:

- интегрирани среди за разработка: MS Visual Studio, Eclipse и др.;

- управление на софтуерните конфигурации: MS VSS, Subversion, CVS, Git;

- автоматизирано тестване: XUnit, JUnit, FITnesse, HP Quality Center и др.;

- непрекъсната интеграция: MSVS Team Foundation, CruiseControl и др.;

- управление на дефектите: Bugzilla, Quality Center, JIRA, TestTrack и др.;

- управление на изисквания: FeaturePlan, CaseComplete, AcceptRequirements, Enterprise Architect, Borland CaliberRM и др.

### 3.3.3. Възможности за използване на виртуални инструментални средства

Виртуализацията е процес, възникнал в резултат на компютъризацията, която се е утвърдила с информатизацията на обществото. Този процес е обективен и характерен за информационните технологии и информационните системи. В контекста на настоящето изследване, под виртуализация разбираме реализация, която осигурява ресурси и услуги, които в действителност не съществуват или не съществуват в начина, по който са представени. Съществуват различни методи и технологии за виртуализация на информационните системи, като виртуализация на хардуера, на софтуерните услуги и комуникации, базирани на интернет (Сълов, 2014; Филипова и др., 2017).

Класическата теория приема системата за управление като единство на два компонента – субект на управление и обект на управление. Тази идея се развива в добавяне на трети компонент, информационната система. Тя заедно със субекта на управление формират системата за управление на организацията. В теорията информационната система се разделя на компоненти, а именно система за икономическа информация основана на информационната база и система за обработка на данни. На свой ред, информационната база се състои от информационен фонд и методи за организацията му (Илиев и др., 2010). На тази основа ние можем да дефинираме виртуалната софтуерна система като софтуерна система, в която са приложени технологии и инструменти за виртуализация, така че информационната база е организирана с виртуални средства и системата за обработка на данни също ползва виртуални среди, технологии, инструменти.

В резултат на виртуализацията се появява нова виртуална организационна форма в бизнеса, наречена бизнес мрежа (б’мрежа). Б’мрежата е обособена система от доставчици, дистрибутори, доставчици на търговски услуги, доставчици на инфраструктура и клиенти, които извършват основна част от бизнес комуникациите и транзакциите си чрез Интернет (Илиев и др., 2010). Според нас, със сигурност, една стартирaща компания като БитПайъниърс може да бъде част от подобна б’мрежа, да ползва предимствата на организацията в тази мрежа за създаване или взаимстване на бизнес модел и създаване на продукт. Част от виртуалните информационни системи и инструменти могат да бъдат също от доставчици на услуги, инфраструктура, търговски услуги и дори клиенти в б’мрежата.

Виртуализацията на информационната система и отделни инструменти може да бъде полезна в няколко насоки, за да се осъществи по-ефективно предприемачески процес.

Като **първи аспект**, най-важно за всяка компания е фокусът към стойността, която добавя виртуализацията към продукта и процеса на неговата разработка. Разработката на софтуер е разработка на интелектуален продукт. За да бъде създаден е нужен интелектуален капитал на компанията, който се състои от човешки капитал, капитал клиенти (потребителски капитал) и структурен капитал (Илиев и др., 2010).

Стартиращата компания има и капитала на предприемачите и техните ресурси в тези три направления. Чрез богатството на достъпни ресурси от данни, знания, мрежи и услуги, които предоставя Интернет, компанията може да увеличи капитала си в трите направления. По отношение на човешкия капитал, тя може да получи знание и информация за желаната сфера, за процеса, за обучение, да наема сътрудници за дадени задачи. За потребителския капитал може да се използват инструменти за проучване на пазар, конкурентно разузнаване, потребители, доставчици, партньори. За структурния капитал, изразяващ се в бизнес процеси и модели за удовлетворяване на пазарни изисквания, може да се проучат съществуващи бизнес модели, отзиви и препоръки за приложението им. Увеличението на интелектуалния капитал е предпоставка за успех в иновационната дейност и то може да се постигне чрез управление на знанието и споделяне на знание. За това важна роля играе приложението и използването на информационни технологии подпомагащи тази дейност.

От казаното следва, че капиталът във всичките му форми е важен за технологичната стартираща компания и трябва да се мобилизира за насърчаване на по-висока производителност, която е чрез създаване (иновиране) на знание и споделяне на знания. Споделянето на знания може да се разглежда като организационна иновация, която има потенциала да генерира нови идеи и да развива нови бизнес възможности чрез социализация и учене на знания (Ling, 2012; Lee, 2016).

**Втори аспект** е начинът на фокусиране в своята основна дейност, която се осъществява с организационно-технологичен процес на десагрегация на елементите на веригата на стойността. Взема се решение за елементите, които да се запазят и за елементите, които да се предоставят на други компании с цел оптимизация. С последваща реагрегация се сглобяват елементите, за да се получи стойностно предложение. След това се прилага мултипликация, за да бъде ефективно осъществено. Поради наличието на множество видове модели на е-бизнес взаимодействие (B2B, B2C, B2G, B2E, C2B, E2B, G2B) (Croll, 2013; Laudon, 2017), според нас е подходящо, компаниите като БитПайъниърс да стартират дейността си, използвайки виртуални модели на взаимодействие с партньори и среда. В тези модели конкуренцията между доставчици за нови клиенти и цена на услуга калкулирана според потребление създава прозорци от промоционални услуги за стартираща компания в ранни етапи на дейност.

**Трети аспект** е, че този процес може да се приложи и към софтуерната система на стартиращата компания. Към нея може да се приложи виртуализация, десагрегация на подсистеми и организиране на транзакциите, последваща реагрегация за цялостна информация за бизнес модела и оценка на приноса.

Като **четвърти аспект**, стартиращите компании като БитПайъниърс могат да осъществят дейности от своя предприемачески процес с минимум ресурси. Такива възможности за тях са:

– Да ползват на ниска или нулева цена достъпно знание и бази от данни;

– Да ползват услуги/инструменти от доставчици на ниска цена според потреблението;

– Да ползват безплатни услуги и продукти за началния период на разработка;

– Да имат лесен виртуален достъп до глобални пазари и отдалечени клиенти.

**Пети аспект** е организация на дейността в компанията чрез средства за виртуален офис и е-бизнес. Виртуален офис за организация е офис, съществуващ в компютърно симулирана среда, в който всички типични офис процеси се извършват чрез използването на компютърни системи. При ограниченост на ресурси, той е алтернатива за стартираща организация. Друга важен момент е, че софтуерът и процесът на разработката му могат да се ползват лесно от глобализацията чрез системите за виртуален офис и е-бизнес (Илиев и др., 2010).

Както е известно, е-бизнесът е форма на организация на бизнеса, при която бизнес процесите, обмяната на информация, финансовите транзакции и прочие дейности се автоматизират чрез информационни системи. Той дава възможности както за разширяване на базата от клиенти, доставчици, географски обхват, прилагане на електронна търговия и маркетинг (Sulova, 2018; Vasilev & Kehayova-Stoycheva, 2019), засилване на вертикалната интеграция и други.

Предвид аргументите, изложени по-горе, можем да твърдим, че използването на виртуализация при експлоатацията на софтуерната система може съществено да улесни стартиращата софтуерна компания. Затова считаме за полезно да се изследват и опишат съществуващи виртуални инструменти и информационни системи, които могат да се използват в предприемаческия процес, като се разделят по етапи и по дейности. За всяка дейност посочваме виртуални инструменти, с които тя да се извърши или да бъде подпомогната. Под виртуални инструменти разбираме онези информационни системи, които са реализирани чрез виртуализация и са приложени принципи за виртуален офис (Илиев и др., 2010):

– Работа в среда на Интернет;

– Дислокация на работните места;

– Експлоатация без посредничество на програмисти;

– Процесорна обработка на информацията;

– Достъпност през уеб компонентът браузър и уеб сървър.

Списъкът с инструменти[[4]](#footnote-5), съставен от нас, е примерен и посочва възможност, покритие на задачи, евентуално разнообразие, като няма претенцията да е изчерпателен.

Първата фаза, която разглеждаме, е „Подготовка”. Тя има за цел валидиране на предприемач, обучение в предприемачество и подготовка на офис и среда за работа. Предприемачите трябва да увеличат своят капитал на тази фаза. Става въпрос основно за човешки капитал с подходящи знания и умения и в по-малка степен структурен капитал[[5]](#footnote-6) чрез въвеждане на подходящи дейности (процеси) на комуникация и споделяне на знания с екип, ментори и други предприемачи. Отделните дейности и подходящите инструменти са представени в табл. 3.3. Както е видно, има много достъпни виртуални инструменти за тези дейности на минимална цена.

Таблица 3.3.  
 Връзка между дейности и виртуални инструменти за фаза „Подготовка”.  
(разработка на автора)

| **№** | **Дейност** | **Инструменти** |
| --- | --- | --- |
| 1 | Годност за предприемачество | Онлайн инструменти за (само)оценка:  psychometrictest.org.uk/entrepreneur-test/  queendom.com/tests/access\_page/index.htm?idRegTest=2287, humanmetrics.com/entrepreneur и много други. |
| 2 | Обучение / Тренинг | www.udemy.com/the-lean-startup/ (Eric Ries)  eu.udacity.com/course/how-to-build-a-startup--ep245 , fi.co,  www.coursera.org/lecture/innovative-entrepreneur/lean-startup-mvps-UyZvW, www.startups.co/platform, scu.edu/mobi/about,  www.startups.co/education, www.enterpreneur.com, |
| 3 | Комуникация с други предприемачи, ментори, екип | Професионални социални мрежи: linkedin.com, twitter.com, shapr.com; Средства за комуникация: Е-мейл; Чат и споделяне (Skype, Viber, Slack), видеоконференции (Hangout, GoToMeeting, Skype), отдалечен достъп (GoToMeeting, LogMeIn);  Консултации: your-startup-guru.com, други; |
| 4 | Подготовка на виртуален офис | Виртуален офис: HotOffice, eRoom, InTandem поддържащ екип и документи, комуникация, презентации и други:  Общи офис приложения: Google Suite, Microsoft Office 365  Средства за презентация  Проекти: bitrix24.eu, atlassian.com, trello.com, samepage.io;  Инструменти за е-бизнес, като например онлайн плащания;  Асистенти: zirtual.com, jobs.bg, zaplata.bg, upwork.com  Онлайн хостинг: многообразие от хостинг услуги |
| 5 | Подготовка на инструменти за стартиращи компании | startupstash.com, www.startups.co/platform, eznumbers.com, 10000startups.com, startup-playbook.com, |

След подготовката или паралелно с подготовката, когато тя има обучение, свързано с реална разработка, е фазата „Изобретяване”. В някои методологии тя се нарича „Генериране на идея”, „Design Thinking” и други. На първо място са включени дейности по разучаване или още наричани разузнаване. Ролята им тук е да се увеличи познанието за клиенти и техни проблеми, състояние, възможности, пазарни условия и т.н. Включени са дейности по дефиниране на проблем, генериране на идея, стратегически анализ, избор и представяне на идея.

В табл. 3.4 е представен списък с дейности на фазата „Изобретяване” и съответните инструменти. Трябва да отбележим, че тук покритието за дейността не винаги е пълно и разнообразно. Причините са, че фаза „Изобретяване” е едновременно, и творческа, и има необходимост от обстоен анализ, за който са налични средства за представяне и търсене на информация. Изборът на данни и източници, инструменти, избор на идея и решения остават на отговорност на предприемача. Характерно е, че фазата винаги завършва с необходимост от подготовка за реално представяне на идеята пред различни аудитории – собствен екип, инвеститори, съветници, клиенти. Цел на това представяне е да се намерят ресурси, да се получи обратна връзка и намерят потенциални клиенти.

Таблица 3.4.   
Дейности и виртуални инструменти за фаза „Изобретяване”.  
(разработка на автора)

| **№** | **Дейност** | **Инструменти** |
| --- | --- | --- |
| 1 | Разучаване |  |
| А | Избор на индустрия | Публични източници за пазара и тенденции, годишни отчети (gartner.com, marketresearch.com, idg.com, други). |
| Б | Конкурентно разузнаване | Пазарни анализи: gartner.com, marketresearch.com  Публични регистри, статистика:  trademap.org, investigativedashboard.org  Преса: builtinla.com/2018/01/16/50-la-startups-watch-2018;  Форуми, блогове, сайт, електронни борси;  Социални мрежи: linkedin.com, twitter.com, shapr.com |
| В | Интервюта с клиенти  \* Лични интервюта, но за отдалечен пазар се ползват и виртуални средства. | За отдалечени клиенти и глобален пазар: социални мрежи (linkedin.com, twitter.com, pinterest.com), търсачки за публикации и въпроси (buzzsumo.com, quora.com, craigslist.com) и анкети (docs.google.com/forms, startupstash.com/forms-surveys, aytm.com, jotform.com) |
| 2 | Дефиниране на проблем | Виртуален офис за документи, презентации, бележки; публични източници за реализирани идеи, за формат на описание: fi.co/madlibs, leanstack.com/leancanvas; |
| 3 | Генериране на идея | milanote.com, curator.co, mindmeister.com |
| 4 | Стратегически анализ | Документи във виртуален офис; публични източници за сектора, тенденции и прогнози. Специализирани диаграми: swotanalysis.com, online.visual-paradigm.com, |
| 5 | Избор на идея | Средства не могат да бъдат представени; |
| 6 | Представяне на идея  (пред екип, съветници, инвеститори и клиенти) | office.live.com/start/PowerPoint.aspx, slideshare.com, docs.google.com/presentation, Видео презентация: biteable.com/presentations, knovio.com, moovly.com |

След фаза „Изобретяване” следва фаза „Бизнес модел”. В нея основните задачи са създаване на бизнес модела и неговото пазарно валидиране. Бизнес моделът съдържа концепция за продукта, именувана с различни термини, като: „продукт”, „уникално стойностно предложение” или „минимално функциониращ продукт” (MVP) (Ries, 2011; Maurya, 2012; Alvarez, 2014; Pompermaier et al., 2019; Melegati et al. 2020).

Валидиране на бизнес модела[[6]](#footnote-7) се прави регулярно спрямо определен пазар от потенциални или реални клиенти. Извършват се тестове за валидност, събират се данни от тях с цел анализ, извличане на знания и вземане на решения за корекция в модела и технологичното решение. Затова показателите, които се избират, трябва да са пряко свързани с дейността и да водят пряко до вземане на решения. В литературата за различни бизнес модели са описани различни системи от показатели (Adler, 2011; Blank, 2013; Croll, 2013):

– AARRR за измерване на нужните действия към клиенти на Дейв МакКлър;

– Мотори на растежа (Engines of Growth) на Ерик Рийс;

– Схема за лесно стартиране (Lean Canvas) на Аш Мория;

– Пирамида на растеж (Growth Pyramide) на Шон Елис;

– Фуния за продажби (Long Funnel) – преход между етапи в процес на продажба;

– Платформа за анализ (Lean Analytics Framework) на Крол и Йосковиц.

Изборът и използването им зависи от процеса, но за всички изпитването на бизнес модела се оценява регулярно. Инструменти за тази фаза също са налични и са изведени в табл. 3.5.

Таблица 3.5.  
 Дейности и виртуални инструменти за фаза „Бизнес модел”.  
(разработка на автора)

| **№** | **Дейност** | **Инструмент** |
| --- | --- | --- |
| 1 | Създаване | Проучване на модели, (не)успешни опити, показатели: leanstack.com, leananalyticsbook.com, startups.co,bizplan.com;  Съвети от ментори: clarity.fm, fi.co, професионални социални мрежи и средства за комуникация: таблица 3.5;  Виртуален офис с документи за описание: таблица 3.4;  MVP чрез списък изисквания или прототип: в таблица 3.7; |
| 2 | Тестване  \* инструментите зависят от бизнес модела, подходът за оценка, показатели и инструменти. Дадените са примерни. | Създаване на БД клиенти: mailchimp.com, launchrock.com,  Анализ на уеб трафик: indicative.com, manageengine.com,  analytics.google.com, analytics.angelfishstats.com/, adobe.com/analytics/adobe-analytics.html;  Маркетинг анализ на поведение: clicky.com, kissmetrics.com, shopmetrics.com, retentiongrid.com, adobe.com/marketing-cloud.html, metrilo.com, marketingplatform.google.com;  Анкети: docs.google.com/forms, jotform.com, startupstash.com/forms-surveys, marketingplatform.google.com;  Тестове: adobe.io/apis/experiencecloud/target.html, abtasty.com/, optimizely.com, |
| 3 | Извличане на знание | Обработка на тестови данни със за стандартен анализ: docs.google.com/spreadsheets, tableau.com/products/cloud-bi, sas.com/en\_us/solutions/cloud-computing.html, turbo.net/hub/powerbi (Power BI online), sisense.com; |
| 4 | Корекция | Средства не са посочени. |
| 5 | Представяне | За презентации (виж таблица 3.10); За диаграми: таблица 3.9 и 3.12; |

Може да се отбележи, че има разнообразни виртуални инструменти и източници на данни за тази фаза. Инструменти не са посочени за дейност „Корекция”, защото тя е резултат от анализ на резултати и вземане на решения за промяна.

Дейностите във фаза „Бизнес модел” и следващата фаза „Разработка на софтуер” се изпълняват постъпково с гъвкава методология, последователно или паралелно по избор на предприемача. Изборът зависи от степента на необходимост от прототип, реално или имитационно изпълнение на изисквания за минимално функциониращ продукт (MVP) във времето. Относно разработката на софтуер, използваните инструменти зависят от избран бизнес модел, архитектура на софтуера и технология за реализиране. В повечето случаи продуктите се реализират чрез многослойно архитектурно решение с отделни взаимно свързани слоеве, при което всеки слой има множество компоненти. Съществуват базови платформи за разработка (frameworks) от един доставчик и допълване с компоненти от други доставчици. Приложимо е не само за компоненти, които липсват, но и за такива, които са по-ефективни от стандартно заложените в платформата. Затова виртуалните инструменти може да са „твърдо” свързани с конкретна платформа или да са по-широко приложими.

В табл. 3.6 са представени стандартни основни и спомагателни дейности за разработка на софтуер, описани в литературата (Филипова и др., 2017).

Таблица 3.6.   
Дейности и виртуални инструменти за фаза „Разработка на софтуер”.  
(обобщено и доразвито от автора)

| **№** | **Дейност** | **Инструмент** |
| --- | --- | --- |
| 1 | Анализ | Виртуален офис за документи;  Средства за проектиране (дейност Проектиране)  Описание на изисквания: atlassian.com/software/confluence |
| 2 | Проектиране | Средства за моделиране: lucidchart.com, cacoo.com,  sparxsystems.com/products/procloudserver/, dragon1.com/products/dragon1-connect-edition, products.office.com/en-us/Visio/, axure.com; |
| 3 | Реализация  \* Забележка: множество от средите поддържат интеграция и на други дейности и на широк спектър продукти. | Кодиране: codeanywhere.com/, aws.amazon.com/cloud9/, koding.com/, progress.com/kinvey, azure.microsoft.com/en-us/products/visual-studio-code, aws.amazon.com/lambda, telerik.com/app-development;  Бази данни: azure.microsoft.com/en-us/services/sql-database/, azure.microsoft.com/en-us/product-categories/databases/, aws.amazon.com/products/databases;  Бизнес логика: aws.amazon.com/lambda, azure.microsoft.com/en-us/product-categories/compute/;  Уеб сървър: azure.microsoft.com/en-us/product-categories/web,  Облачни услуги: Доставчиците имат много услуги – от API, изчисления, балансиране, опашки до изкуствен интелект;  Хранилища за модули: nuget.org, npmjs.com, github.com, openwrap.org, incubator.apache.org/npanday,chocolatey.org;  Потребителски интерфейс: angular.io, reactjs.org, vuejs.org, startbootstrap.com, mockplus.com, jqueryui.com, jqwidgets.com, riot.js.org, semantic-ui.com, webix.com; |
| 4 | Тестване – Верификация и Валидация | Автоматизирана интеграция и инсталиране на тестови среди: github.com, jenkins.io, gitlab.com, circleci.com, cloudbees.com, azure.microsoft.com/en-us/services/devops/;  Системи за тестове: gurock.com/testrail, seleniumhq.org, techexcel.com/products/devtest/, getxray.app, ranorex.com, getzephyr.com;  Автоматизирани тестове: ranorex.com, https://percy.io/, Сътрудници тестери: utest.com/about-us, |
| 6 | Внедряване | Средства за автоматизирано инсталиране: jenkins.io, azure.microsoft.com, www.gitlab.com, cloudbees.com, ; |
| 7 | Поддръжка | atlassian.com/software/jira/service-desk, atlassian.com/software/jira/ops, |
| 8 | Управление на проекта | azure.microsoft.com/en-us/services/devops,  atlassian.com/software/jira, atlassian.com/software/trello,  github.com/features/project-management, gitlab.com, |
| 9 | Управление на конфигурации и промени | gitlab.com, atlassian.com/software/crucible, |
| 10 | Конфигуриране на среда за работа | Избор на инструменти и настройка: azure.microsoft.com,  Среди за програмиране, управление на проект (погоре);  Хранилище: github.com, atlassian.com/software/bitbucket |

Включени са примерни виртуални инструменти за разработка на многослойно уеб приложение, основно с технологии и продукти, свързани с облачните платформи за разработка на компаниите Microsoft (Azure) и Amazon (AWS), като може да използват възможностите, които се предлагат и от други доставчици на облачни услуги (напр. Google, Alibaba Cloud, IBM Cloud, Oracle Cloud, Salesforce и др.). От резултатите е видно, че съществува голямо разнообразие от доставчици и средства, и покритие на дейности. Популярни в практиката са интеграции между инструменти. Според нас, особен интерес представляват три модела за интеграция между инструменти:

– Основният продукт е настолен, но се интегрира с много виртуални инструменти (Пример Microsoft Visual Studio с пакети и облачни услуги);

– Заложена интеграция чрез събитие от друг инструмент;

– Интеграция на инструменти чрез допълнително частно разработена обработка на събития.

В обобщение на представеното, и както е видно от табл. 3.3, 3.4, 3.5 и 3.6, за почти всяка дейност са налични виртуални инструменти, предоставяни като услуги. Въпрос на преценка на предприемача е кои от тях и доколко са удобни за даден продукт, идея и бизнес модел. По отношение на цената за използване, повечето инструменти имат нулева или ниска цена – безплатни версии, безплатни за микро екипи (до 5-10 потребителя) или цената е според брой потребители или обем консумация, което представлява голямо улеснение за една стартираща компания като БитПайъниърс, тъй като предполага ниски разходи за употреба. Съществен проблем е многообразието от инструменти за употреба и изборът на подходящ инструмент. В противовес на това, множество от продуктите имат функции, които покриват повече от една дейност и/или могат да се интегрират с трети инструменти за други дейности, което е улеснение за изграждане на една интегрирана среда за работа в стартираща компания.

От изложените резултати от проведените изследвания за виртуализация, можем да направим обобщение, доколко дейностите в предприемаческия процес могат да се извършат с виртуални инструменти. Според нас, е необходимо да се съобразим с два основни показателя – покритие и разнообразие. Първият показател дава отговор на въпроса, доколко за задачите в дейността са разработени виртуални инструменти. Вторият показател – разнообразие, представлява степен на възможен избор от различни достъпни инструменти. За тяхно представяне използваме относителна скала с числа от 0 до 3, която е дадена в табл. 3.7, като синтез от двата показателя. Предлаганата от нас скала не отразява съотношение, а по-скоро групиране по тези два показателя с цел по-ясно отразяване и представяне на степента на покритие на задачи по дейности.

Таблица 3.7.  
 Скала на покритие на дейност с виртуални инструменти.  
(разработка на автора)

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Разнообразие от различни достъпни инструменти* | | |
| **Покритие на дейността от налични виртуални инструменти** | *Ниско* | *Средно* | *Високо* |
| **Липсва** | 0 | 0 | 0 |
| **Ниско** | 1 | 1 | 2 |
| **Средно** | 1 | 2 | 3 |
| **Високо/Пълно** | 2 | 3 | 3 |

Следвайки дадените показатели и дадената скала в табл. 3.7, можем да обобщим резултатите за покритие на дейностите от разглежданите инструменти (табл. 3.8).

Таблица 3.8.  
 Покритие на дейности по фази с виртуални инструменти.  
(разработка на автора)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Подготовка​** | **Изобретяване​** | **Бизнес модел​** | **Разработка на софтуер​** | |
| Годност​ | Разучаване​ | Създаване​ | Анализ​ | Проект |
| 2 | 2 | 2 | 2 | 2 |
| Обучение​ | Проблем​ | Тестване​ | Проектиране​ | Конфигурации​ |
| 2 | 1 | 3 | 2 | 2 |
| Комуникация​ | Идея​ | Знание​ | Реализация​ | Среда |
| 3 | 1 | 2 | 3 | 3 |
| Вирт. офис | Стратегич. анализ​ | Корекция​ | Тестване | Внедряване​ |
| 3 | 1 | 0 | 3 | 2 |
| Среда | Избор на идея​ | Представяне​ | Валидация​ | Поддръжка​ |
| 1 | 0 | 2 | 2 | 2 |

В табл. 3.8 са включени общо 25 дейности. Дейност „Представяне на идея” и „Представяне на бизнес модел” са обединени, поради покритие на инструментите. Както е видно от таблицата, за две дейности няма посочени инструменти, а 4 дейности са с ниско, 13 са със средно и 6 с високо покритие и разнообразие. Данните от таблицата показват, че за 75% от дейностите имат налични виртуални инструменти със средно или високо покритие, и разнообразие. За останалите 25%, или липсват виртуални инструменти, или разнообразието е малко. Това ни дава основание да потвърдим в значима степен нашето първоначално предположение, че с виртуални инструменти могат да се изпълняват голяма част от задачите по разработката на нов продукт в стартираща софтуерна компания. Най-ниско е покритието във фаза „Изобретяване” и най-високо в „Разработка на софтуер”. Определено се забелязват възможности за иновация в инструментите за стартиращи компании в две посоки – увеличаване на покритието на задачи по дейности и интеграция в единна среда.

***Изводи и обобщения към трета глава***

1. При използването на система за управление разработката на софтуер в технологична стартираща компания е необходимо да се избере и подходящ подход за разработка. Прилагането на подходящ подход за управление на процесите на разработка очакваме да осигури постигането на висока ефективност на работа и да се окаже съществен фактор за успех на проекта, по който се работи. Поради тази причина изборът на подход при разработката на софтуер е също толкова важен, колкото е и използваната софтуерна система, прилагана за автоматизиране на отделните дейности по разработката на софтуер.

2. Разработен е план за реализация, внедряване и експлоатация. Един от основните моменти при разработката на софтуерна система е планът за реализация, който включва както планиране на реализациите и версиите, така и характеристики, които се реализират в тях, заедно с разпределение във времето. Разработен е съкратен вариант на работния продукт на база сценарии за взаимодействие. При необходимост той може да се разширява и допълва според потребностите и стратегическите виждания на ръководството.

3. Производството на софтуерни продукти е специфична дейност и в зависимост от ситуацията е подходящ точно определен подход. Според нас, в повечето случаи за разглеждания обект на приложение е подходящо прилагането на гъвкав подход за управление на софтуерните проекти, базиран на подход CC и полезни добавки от SCRUM, LSD, UP, DSDM, XP и др. Предложени са няколко варианта – лек за малки (клиентски) проекти и услуги, и нормален за по-големи клиентски проекти и собствени продукти.

4. Използването на виртуални инструментални средства при експлоатацията на софтуерната система може съществено да улесни стартиращата софтуерна компания. С тази цел са изследвани и систематизирани съществуващи виртуални инструменти и информационни подсистеми, които могат да се използват в предприемаческия процес, по отделни етапи и дейности. За всяка дейност са посочени виртуални инструменти, с които тя може да се извърши или да бъде подпомогната.

# Заключение

Настоящият дисертационен труд разглежда въпроси, свързани с

1. Имат се предвид подходите, базирани на Manifesto for Agile Software Development, https://agilemanifesto.org/ [↑](#footnote-ref-2)
2. По лични наблюдения, опит и мнение на автора. [↑](#footnote-ref-3)
3. Project Management Institute подход с правила за работа по проекти на база стандартен дефиниран процес. [↑](#footnote-ref-4)
4. По наше мнение, популярни в практиката системи с общо предназначение като виртуални инструменти към 2021 г. са: Jira (https://www.atlassian.com/jira) , Basecamp (https://basecamp.com/), Zoho (https://www.zoho.com/), Trello (https://trello.com/), Asana (https://asana.com/), Podio (https://podio.com/), Todoist (https://todoist.com/) и др. [↑](#footnote-ref-5)
5. Под „структурен капитал“ имаме предвид ИТ и другата инфраструктура, която подпомага човешкия капитал в работата му. [↑](#footnote-ref-6)
6. В случая имаме предвид процеса на установяване дали избрания целеви пазар има потребност от разработвания софтуерен продукт. [↑](#footnote-ref-7)