# Глава 2. Архитектура на облачна система за управление на поръчките от клиенти

## 2.1. Същност, цел и обхват на софтуерната архитектура

Думата „**архитектура**“ често се използва в контекста на нещо от високо ниво, което е отделено от детайлите на по-ниско ниво (Martin et al., 2017). Софтуерният продукт, разглеждан в настоящия труд, ще се състои от 2 клиентски приложения, които ще се свързват към клъстер от услуги.

### 2.1.1. **Ключови бизнес процеси и дейности при системата за** управление на поръчките

Тази подточка ще представи важни случаи на употреба, които са критични за бизнеса, част от основния домейн. Ще бъдат използвани диаграми на бизнес сценариите. Те ще идентифицират действия, които очакваме потребителите да направят.

- Обхват на мобилното приложение



- Обхват на уеб портала

-Обхват на с

https://blog.openreplay.com/jwt-authentication-best-practices

.

### 2.1.2. Информационно бизнес моделиране на системата

Така представеният обхват от желана функционалност представлява концептуалната рамка на софтуерната система. По отношение на проектирането приемаме за съществени компоненти следните дейности: детайлизиране на изисквания, анализ и проектиране на информационен модел. Тъй като системата се базира на работа с потребители, според нас е подходящо да се използва архитектурно центриран модел, базиран на сценарии за взаимодействие. Затова е разработен бизнес модел (БМ) на системата и се придържаме към утвърден подход чрез използване на профил за информационно бизнес моделиране (ИБМ), базиран на универсалния език за моделиране и описание на софтуерни системи UML като признат международен стандарт за обектно-ориентиран език за модели[[1]](#footnote-2). За целите на ИБМ, според нас, е подходящо да се използва UML с разширение чрез специализация с профил на IBM RUP за бизнес моделиране (Fellmann et al., 2018; Beynon-Davies, 2018; Musulin & Strahonja, 2018; Sadowska & Huzar, 2019; Wirtz, 2019). Следвайки този подход, може да се откроят следните елементи за софтуерната система:

1. **Бизнес сценарии за взаимодействие със системата (БСВС)** – отразяват цел и обхват на бъдещата софтуерна система, взаимодействие с потребители и какви услуги осигурява.

2. **Концептуален бизнес модел (КцБМ)** – показва начина, по който служители и обекти, които те управляват, се свързват (статично и динамично), за да се реализират съответните БСВС.

3. **Обектен бизнес модел (ОБМ)** – част от КцБМ, отразяваща само бизнес същности (характеристиките и връзките им) и не включва отговорности на бизнес работниците.

При разработката на модела се ограничаваме единствено над основния БСВС. В него участници са описаните по-горе роли потребители, а описанието на **бизнес сценария за взаимодействие със системата** относно разработката на софтуер е в следните стъпки:

1. Описание на продукт – вътрешен възложител, координатор или главен проектант описва продукта или негова версия с неговата визия.

2. Дефиниране на проект – вътрешен възложител или координатор дефинират проект, условия, ограничения, характеристики и изисквания. Те също дефинират шаблоните, които ще се ползват в проекта за процес, оценка и т.н. Възложител или потребител дефинира списък със сценарии или характеристики.

3. Планиране на проект – възложител ранжира характеристиките; главен проектант и екипа[[2]](#footnote-3) определят вложения труд за тях и ги съобразяват с капацитета. Получава се оферта и се дефинира договор/план с параметри.

4. Организиране на разработка – главен проектант/координатор дефинира процес по избран шаблон, екип, етапи, реализации. С помощта на екипа се дефинират първи итерации и задачи, списък рискове, и се разпределя времето.

5. Планиране на реализация – главен проектант/координатор с възложител актуализират плана с реализации, разпределят характеристики по реализации, добавят и оценяват нови характеристики. Екипът описва итерации, задачи и преценки за тях, списък с рискове, необходима документация и други.

6. Работа по задачи – екип на компанията работи по задачи, ползва лично табло за статус, променя статус на задачи, отчита разход на време и средства, дефинира нови задачи, проблеми и дефекти, поема и обработва дефекти и рискове. Ежедневно записва изразходвано и оставащо време за работа по задачи, и промяна на статус. Преглежда справки за статус и очаквано завършване.

7. Приключване на итерация – координатор/екип отчита резултатите и записва изводи от отразяващото подобрение. Модифицира се плана за следващи итерации и задачи, преглежда справки за статус и очаквано завършване. Екипът коригира задачи и оценки в следващи итерации (координатор в реализации).

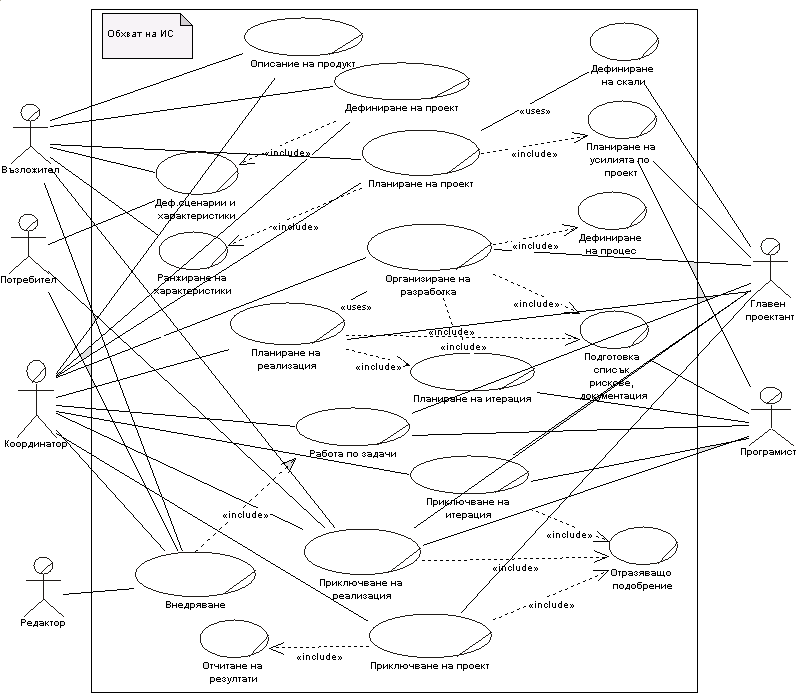
8. Приключване на реализация – координатор/екип отчита резултатите, оформя продукт, преглежда справки за статус, дефинира пречки, дефекти, рискове и модифицира план за реализации след обсъждане с възложител/потребители; координаторът записва резултати от отразяващо подобрение, а екипът дефинира задачи за следващи итерации. Коригира се документацията.

9. Внедряване – главният проектант/програмистите изпълняват задачи за финални корекции и дефекти, внедряване и приемни тестове. Програмистите оформят версия и внедряват. Редактор оформя документацията.

10. Приключване на проект – координатор/екип отчита резултатите, правят се отчети за ефективност и финансов резултат от проекта. Дефинират се резултати от отразяващо подобрение. Дефинират се задачи с лична отговорност в друг проект за подобрение относно подход/конвенции за разработка.

11. Общи административни задачи – координаторът работи по общи административни задачи, настройва системата, подготвя информационни табла и други.

Така описаният главен сценарий е показан на фиг. 2.1[[3]](#footnote-4). Той описва взаимодействието между бизнес актьори и софтуерната система при разработка на софтуер. Сценариите са представени с овални форми. По-големите са сценариите в главния БСВС, а по-малките са някои интересни детайлизации към основните. Връзките между актьори и сценарии показват извършваните действия. Връзката „uses” показва, че един сценарий използва друг, а „include” показва, че един сценарий включва като част от себе си друг сценарий.



Фиг. 2.1. Диаграма на главен бизнес сценарий за взаимодействие „Разработка на продукт”. (разработка на автора)

Главният сценарий относно поддръжка на софтуерен продукт е следния:

1. Дефиниране на поддръжка – координатор/главен проектант дефинира изисквания, условия, мониторинг при поддръжката;

2. Следене на заявки – възложител/координатор попълва заявки от клиент; програмист извършва мониторинг и попълва заявка за некоректен резултат;

3. Конвертиране на заявки – заявките се преобразуват в дефекти, задачи или характеристики за обновяване на продукт и се назначават за програмист;

4. Опис на внедряване – координатор/програмист описва внедрени версии;

5. Отчет за поддръжка – координатор/възложител/потребител описва мнение за удовлетвореност от внедрена версия и оказана поддръжка.

Главните сценарии може допълнително да се детайлизират в подчинени БСВС. По отношение на **концептуалния бизнес модел** (КцБМ), в него се изгражда обектен модел, представящ бизнес същностите с информационен характер. Бизнес същностите и основните им свойства са следните:

1. Продукт – разработван продукт с цел, описание и списък с версии по вид и номер.

2. Продуктова версия – версия на даден продукт с вид, номер и кратко описание.

3. Проект – разработван проект за даден продукт; описва се име, код, въведение, цел, идентификатор на продукт и версия, условия, ограничения, особености, целеви срок и бюджет.

4. Списък сценарии – работен продукт като „Actor-Goal List” или „Use Cases”, който описва действията – актьор, име, цел, стъпки, автор, пореден номер.

5. Характеристика – изискване за продукт/проект с йерархична структура, теми, статус за актуалност/изпълнение, ранг за стойност и риск, приоритет, оценки за реализации.

5. Клиент – външен клиент възложител.

6. Представител на клиент – лице представител на клиент с роля в проекта.

7. Проектен план – план/оферта/договор с условия, ограничения, качество, цена, срок, планирани усилия за характеристики и параметри за „конус на несигурност” (резерви).

8. Служител – член на екипа в технологична стартираща компания; име, история и текущи данни за позиция, умения, ставка, документация, параметри за лична преценка на усилията, роля(и) в проекта.

9. Участник – член на екипа (служител или представител на клиент) с роли в проекта.

10. Роля – именувана роля за участник в проекта – определя персонализация и достъп.

11. Екипна структура – назначение на участник в проект и роля/роли.

12. Етап – етап от проект – име, вид, статус, планов/актуален срок (дължина, от-до).

13. Реализация – вид етап, който притежава описание на реализирания в края продукт.

14. Реализирано приложение – версия на приложение реализирано към клиент.

15. План за реализация – схема на етапи и реализации (със срок) в проекта със списък характеристики подредени за ранг/зависимост и списък реализирани приложения.

16. Оценка за усилия – оценка за усилията на член на екипа по дадена характеристика/работа. Дава се в единици от мярка за най-вероятен и сигурен.

17. Списък рискове – работен продукт за проект описващ рискове за дадена реализация и евентуално характеристика, подреден по приоритет. Рисковете са с наименование, ранг в реализация, вероятност по скала, опис на вероятен резултат, статус и връзка с елемент.

18. Итерация – итерация в дадена реализация, със срок (от, интервал от време, до), списък със задачи.

19. План за итерация – работен продукт със задачи и други елементи в итерация.

20. Задача – работа за реализация на характеристика от определен участник; има име, описание, статус, начална и текуща оценка, оценка за оставащо време за завършване.

21. Тест – приемен тест като подвид задача, която да се изпълни в дадена итерация.

22. Проблем – описан проблем в работата на софтуера – има заглавие, характеристика, тип, статус, приоритет, контекст, вход, изход, автор, отговорник и други.

23. Дефект – вид задача създадена от проблем за поправяне със статус, модул, решение.

24. Заявка – заявка/идея за продукта в даден проект, която има име, тип и описание и може да бъде превърната по-късно в нова характеристика, дефект, задача или отпада.

25. Подобрение – заключение от отразяващото подобрение към реализация/итерация. Има списък мерки с тип, описание, автор и отговорник. Може да се превърне в задача от проект.

26. Оставащо време – очаквано време на член от екипа за работа до края на задача.

27. Разход на време – отчитане на дадена дата, период от време, времето за което член от екипа е работил по дадена задача (за една характеристика) и коментар по това.

28. Календар – разполагаемо работно време в часове по дни на различни участници.

29. Финансов разход – отчита кога, кой, какъв тип, за какво и колко разход е направил.

30. Записка – бележка/съобщение от софтуерната система или участник за събитие или друго относно проект или друга същност. Има тип, заглавие, съобщение, време, статус, уведомление. Може да се ползват и като бележки за дизайн от автора на документ (работен продукт).

31. Документ – документ на външен файл, с който елемент от системата има връзка.

32. Таг – списък от индивидуално дефинирани и именувани стойности за един обект[[4]](#footnote-5).

33. Рангова скала – за подреждане на обекти за стойност[[5]](#footnote-6), риск, вид, приоритет и т.н. Скалата има списък опции със свойства като код, име, ред, тежест, коментар.

34. Процесен шаблон – именуван и описан списък с етапи за даден тип процес и дефиниция на скалите, които се ползват. Етапите имат име, дължина на срок и описание.

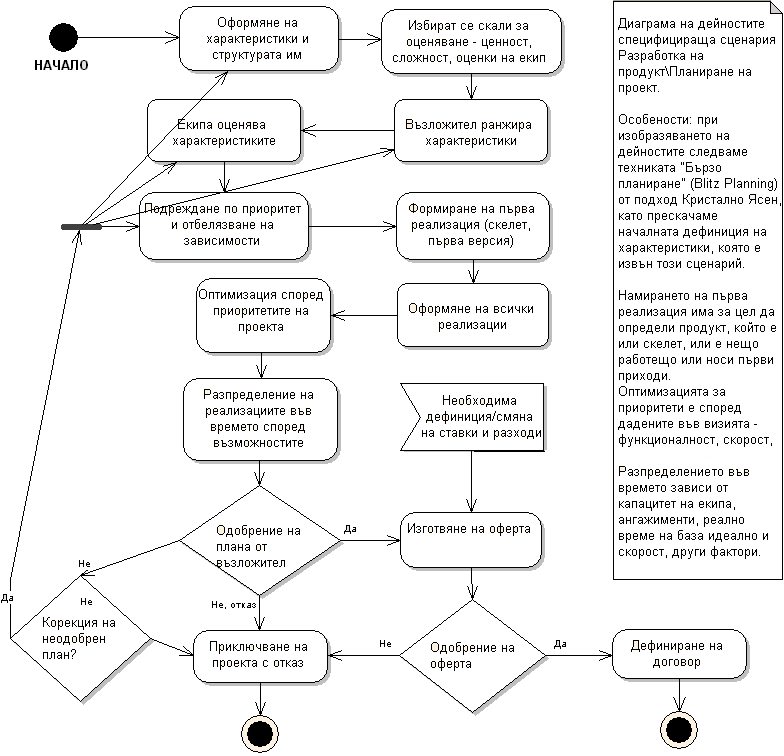
В компонента „поддръжка на продукт” са налични множество сходни дефиниции на същности както в компонента „разработка на продукт”. Характеристиките са изисквания в опростен линеен списък с теми, заявка и тест, и имат някои разлики. Различни **бизнес същности** са следните:

– Мониторингов тест – тест за регулярна проверка на продукта;

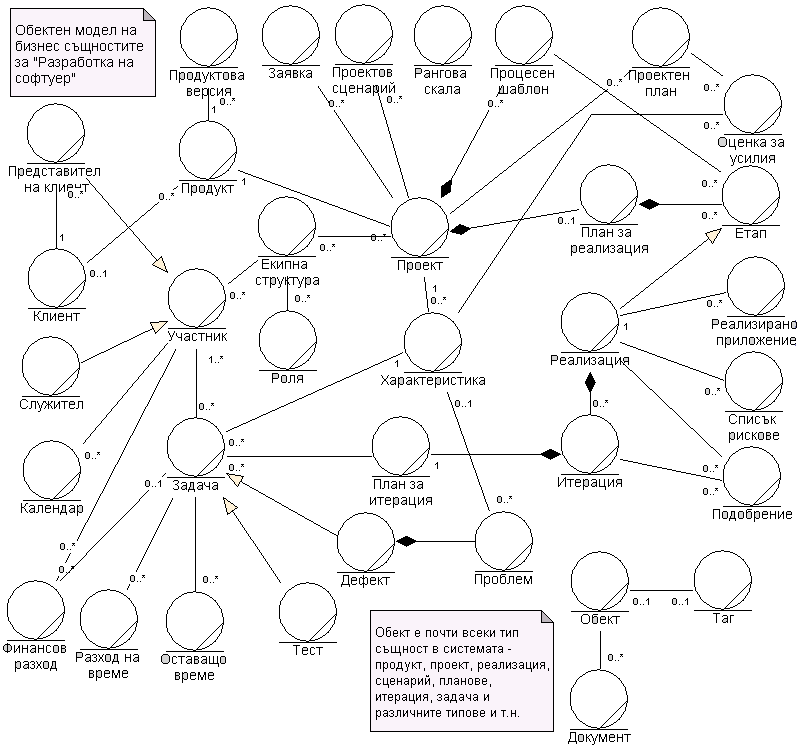
– Заявка за поддръжка – заявка за необходима промяна/корекция/проверка за нередност – може да се конвертира в задача/характеристика/дефект.

По отношение на **реализацията на БСВС** са разработени диаграми на взаимодействията (на последователност) и колаборативни диаграми, с които се показва взаимодействието между бизнес работници (с роли в технологичната стартираща компания – координатор, главен проектант, програмист, редактор, тестер), и рамките на техните отговорности. Техните отговорности са основа за формиране на функциите на системата и персонализираният достъп за всяка роля. Бизнес актьорите са външните участници – възложител и потребител.

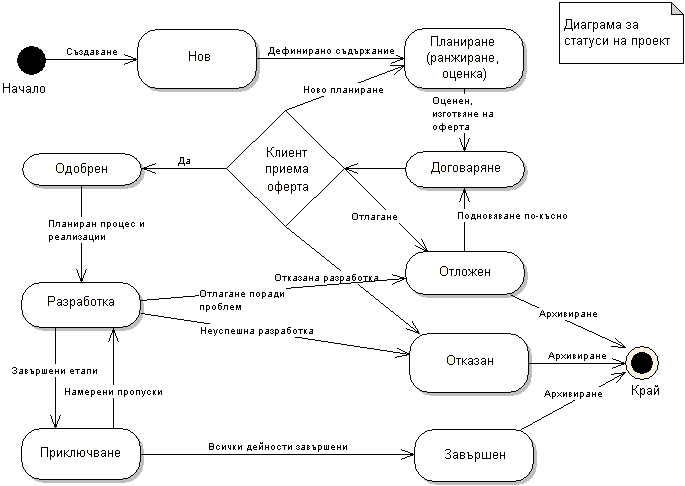
На фиг. 2.2 са представени детайлите в отделни бизнес сценарии с диаграма на дейностите и е показано описването с дейности на даден сценарий. Детайлното представяне на бизнес същностите в обектния модел е дадено на диаграмата на фиг. 2.3. Тя показва и връзките между тях. За да се отразят състоянията на тези бизнес същности и събития, при които те се променят, е използвана диаграма на състоянията. Примери относно компонентите „състояния за проект” и „реализация” са дадени на фиг. 2.4 и фиг. 2.5 чрез диаграма за възможни състояния на същност „Проект” и събитията, при които тези състояния могат се променят. Тези диаграми са основа и за изработване на списъка от статуси за дадения обект, както и логиката, която е залегнала в приложението за възможните преходи.



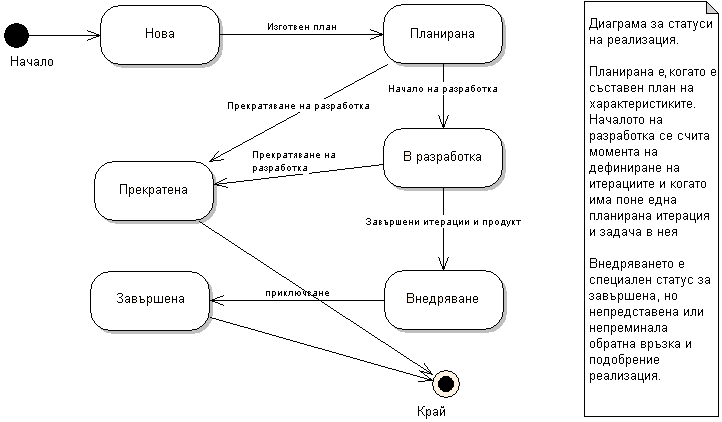
Фиг. 2.2. Диаграма на дейности за БСВС „Разработка на продукт”/”Планиране на проект”. (разработка на автора)



Фиг. 2.3. Обектен модел на бизнес същностите „Разработка на продукт” – диаграма на бизнес същности и техните връзки с други бизнес същности. (разработка на автора)



Фиг. 2.4. Диаграми на състоянията на бизнес същност „Проект”. (разработка на автора)



Фиг. 2.5. Диаграми на състоянията на бизнес същност „Реализация” (разработка на автора)

В съответствие с изложения в първа глава обзор на текущото състояние на информационните системи – тенденции за групова работа (чрез система съобразно отговорности), централизация на данни и достъп, бездокументални и интернет технологии, приемаме за изграждане на софтуерната система да се използва технология тип „клиент-сървър”. Всеки участник в процеса работи на софтуерен клиент, а базата от данни, логиката и обработката се извършват основно на един или няколко сървъра. Според нас, в случая трислойната архитектура за реализирането й е подходяща – представителен слой за потребителски интерфейс, слой за бизнес логика и слой за данни (Microsoft Patterns & Practices Team, 2009; IBM Cloud Education, 2020). В този случай на трислойна структура, софтуерната система е необходимо да се структурира в отделни компоненти, които са разположени в отделните слоеве. Детайлизация на компонентите е дадена по-нататък в изложението, като е описана композицията при реализация на софтуерната система. Реалният потребител – член на екипа, контактува само с представителния слой чрез входно-изходен интерфейс на уеб приложение.

### 2.1.3. Възможности за усъвършенстване на информационната база

Изхождайки от принципа, че софтуерната система е средство, с което може да работи паралелно целия екип, според нас, е необходимо да се заложи възможност за участие и достъп на клиент (негов представител) до софтуерната система. С директен уеб достъп възложител или потребител може да види реализираните версии и характеристики, готови за тестване, някои отчети и прогнози за завършване, и т.н. Адаптацията на предложения подход води до спестяване на време за комуникация на екипа в технологичната стартираща компания. При експортно ориентираните технологични стартиращи компании е от полза и наличието на многоезикова поддръжка.

В проекта за софтуерна система е необходимо да се предвидят и автоматизация на процеси: първо, да притежава възможност за дефиниране на процес от шаблон, и второ, да са заложени правила за автоматична промяна на статус и уведомления. Правилата може да се поставят за характеристики, етапи, рискове и други, в случай, че за тях се дефинират свързани елементи – задачи, дефекти и т.н., или всички такива бъдат приключени. На следващ етап на разработка в перспектива софтуерната система е възможно да се добави функция за автоматичен начален избор на характеристики по реализации на база приоритет в стойността/риска, оценки за тях, процес и срок на проекта, и капацитета работно време.

По отношение на **пренасянето на данни и съхранението**, следва да се използват общоприети стандарти и мрежови протоколи за комуникация, както наложили се в практиката системи за управление на бази данни (СУБД) – релационни, така и от тип NoSQL. За съхраняване на работните продукти е подходящо да се разчита на външно хранилище на документи – публична или частна облачна услуга, или локална система за управление на конфигурациите, която да пази историята на промените. Възможно е прилагането и на хибриден подход. Перспективно направление е системата да има възможност за индивидуализация и промяна на списъците за подреждане (ранжиране) и шаблони за процеси. Така технологичната стартираща компания може да изработи собствени варианти на процес. За правилното им прилагане потребителят може да използва заложените в софтуерната система инструкции за употреба на шаблони и скали.

По отношение на **интегриране с други системи** – в практиката минималните очаквания на потребителите са да е предвиден експорт на данни от отчети и списъци във формат на Excel таблици. Чрез възможност за програмен достъп до библиотеки/услуги, тази функционалност може да се интегрира с външни средства. Например средства за управление на изисквания, за тестване, среди за разработка (така че работните елементи от софтуерната система да се появят в средата за разработка на програмиста, а също и от там да се получат междинни данни за извършвана или завършена работа). Друг пример е онлайн система за заявки за поддръжка, промени и дефекти, (автоматизирано) публикуване на отчети в сайт на компанията, или интеграция с външен софтуер за управление на изисквания, управление на конфигурации, управление и отчитане на време и т.н. Така дефинираните свойства се използват/попълват, а липсващите в системата се добавят в тага в една избрана структура – XML или именувани двойки.

По отношение на **потребителския интерфейс** съществува тенденция, при съвременните бизнес информационни системи, той да е лесен за употреба и гъвкав (лесен за настройване). Поради високата динамика на смяна на устройства и парадигми за дизайн считаме, че в началния етап на развитие на софтуерната система не е удачно да се прилага високо качество на интерфейса, каквато е практиката при масовите продукти. Достатъчни е да се ползват базови възможности на средата за разработка за удобен интерфейс.

Като се отчита, че технологичната стартираща компания има малък екип, ниска формализация на подход и сложност на проекти, е удачно интерфейсът да притежава следните параметри:

– малък размер – оптимизиран за малко данни[[6]](#footnote-7)/съдържание, като избягва страниране;

– простота – опростени данни/съдържание, кратко, най-необходимото, без сложност;

– стандартен формат – стандартен формат на работната площ и еднотипни екрани;

– меню с две нива – постоянно главно меню и подменю за избрана опция;

– лично табло – за секция; персонализирано; има справки/връзки за най-важното;

– лесна трансформация – лесна трансформация на елементи и смяна на статус/ред[[7]](#footnote-8);

– лесен достъп до инструкции за опции, статуси, рангове;

– лесно прикрепяне на записки – записки, таг, документи лесно се задават за обект;

– избягване на мултизадачна работа – фокус над един проект/задача в един момент;

– бързо и лесно отчитане на работа и време;

– автоматичен работен филтър – по подразбиране се показват само работни елементи;

– синхронизация при екипна работа – регулярно интерфейса да е синхронизиран;

За приложението, което се достъпва от потребителите, е удачно да се използва един основен макет на интерфейс с няколко статични елемента, главно статично меню, подчинено меню, работна площ за повечето екрани и списъци с бързи връзки за най-важните, спешни и нови елементи свързани с текущите продукт, проект, функция и потребител. Детайлно този въпрос е развит в по-нататък в изложението.

Първичното отчитане в настоящата софтуерна система е ръчно от участниците в проекта. Данни се въвеждат във входни екрани от участниците лично и ръчно. На първоначалните етапи от функционирането си софтуерната система може да не предлага автоматична отчетност, както е например в eScrum (Newmark et al. 2018; Dixit & Bhushan, 2019). Това се компенсира с простота и удобство при въвеждането на данни. Също така може да има известна компенсация за отчитане на времето, ако се попълва лога за отчет на работното време със записки, които показват кой и кога е работил със системата и така по-лесно член на екипа може да си спомни и укаже по какъв проект и задача, кога и какво е работил.

Много от екраните за личен отчет трябва да могат да работят както с отговорен потребител, така и с координатора, за да може програмистите да са заети с работата си по същество и с по-малко административна работа. Удачно е извеждането да е само в електронен формат – на екран, експорт в HTML уеб страница или данни в Excel-файл. Изключения може да има в следните наложителни случаи:

– интеграция с външни продукти, които да правят електронен трансфер на данни;

– разпечатване на документи за възложителя, където се изисква писмена форма (оферта в търг, приложение към договор и т.н.);

– разпечатването на документи за т.нар. „информационни радиатори” съгласно подхода.

Предложения концептуален модел на софтуерна система е опростен за целите на технологичните стартиращи компании. Той подпомага компанията за ефективното управление и изпълнение на нейните проекти. Вниманието при софтуерната система се концентрира над разработката на проекти във връзка с разработката на продукти.

Функционалността на системата се различава от готовите продукти на пазара в следните направления (пропускаме разликите в базовите подходи и стриктното предназначение на предлаганата софтуерна система за технологични стартиращи компании):

– опростява работата със системата, като се избягва сложността на готови продукти като „VersionOne” (https://www.collab.net/products/versionone) и eScrum (https://www.digibelt.com/en/escrum-prod-5);

– акцентира се върху асинхронното, а не върху синхронизирано управление на множество проекти и екипи („VersionOne”, и др.);

– зависимостта между задачи е премахната – считаме това и по-горното за излишна сложност[[8]](#footnote-9);

– важно предимство е простота на екраните, които наблягат на спецификата за малък обем и фокус;

– в резултат на опростяването на функционалността е избегнато използването на „работна площ” с влачене и синхронизирани екрани;

– вградено рефлексивно подобрение и мерки по отговорно следене за изпълнението им в същия или отделен проект;

– техническа възможност за интеграция с външни продукти чрез външен програмен достъп (но на този етап няма интеграция с конкретен външен софтуер или среда за разработка);

– опростени средства за прогнозиране на очакван край, като алтернатива на обширните отчети;

– отчет за ефективност както в ScrumWorks (https://www.collab.net/community/scrumworks);

– вградено е поддържане на данни за човешки ресурси и следене на промените при тях;

– изработка на оферта/план, което при други системи може да липсва;

– специална секция за продуктово следене;

– съпровождане на продукти (което имат само част от горните примери).

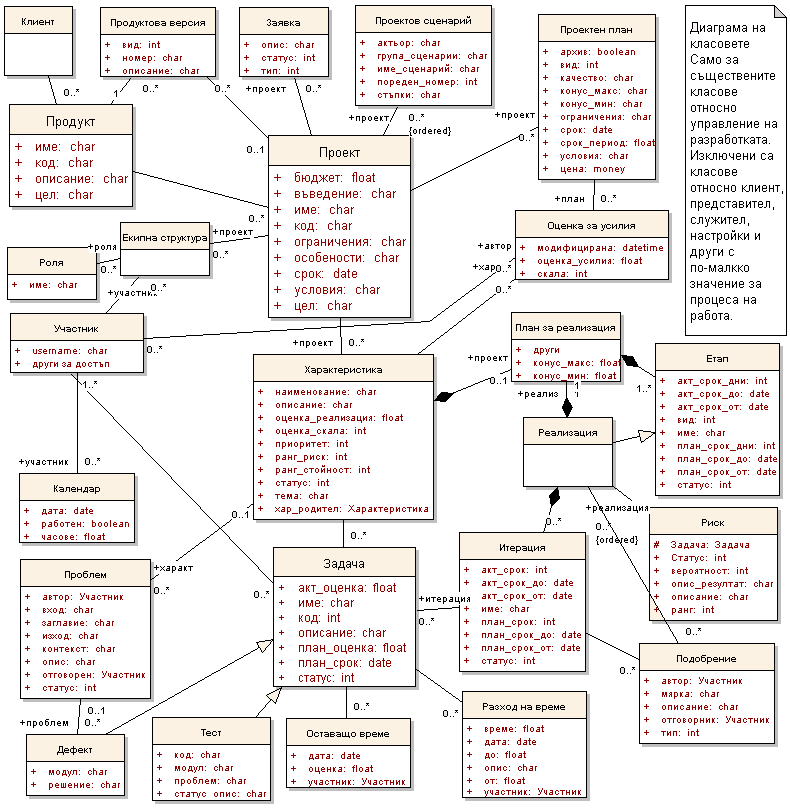
Предлаганият концептуален модел на софтуерна система се фокусира над разработката на софтуерни проекти във връзка с разработката на продукти, прилагайки гъвкав подход (подход за лесна адаптация към промените). Той е опростен за целите на технологичните стартиращи компании и очакваме да подпомогне компанията за ефективното управление и изпълнение на нейните софтуерни проекти.

## 2.2. Логически модел на системата

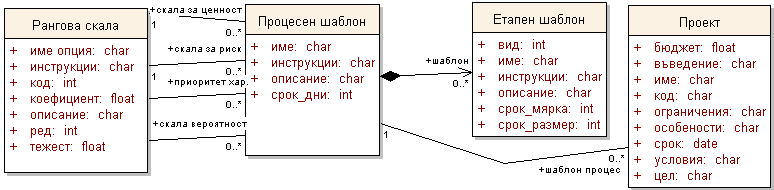
Логическият модел е разработен на базата на изложения концептуален бизнес модел в концепцията за софтуерната система. За постигането на целта, в началото следва да се обособят и развият бизнес същностите като класове със съответни свойства и връзки.

### 2.2.1. Диаграми на класове, реализиращи бизнес същностите

Диаграма на основните класове и техните връзки са показани на фиг. 2.6 – съществените класове за разработка, формирани от бизнес същностите, и фиг. 2.7. Под основни разбираме тези класове, които се отнасят пряко към управление на разработката. Изключени са детайли по служители, клиенти, договори, поддръжка и т.н. Добавена е диаграма за класовете, обслужващи шаблони за процеси и индивидуални скали за ценност, риск, вероятност, степен и други. Компонентите в нея помагат за настройката на предварителни макети за различни процеси. Като цяло тези и сходни диаграми са и основа за разработване на база данни за софтуерната система и скелет за класовете в бизнес логиката на софтуерната система.



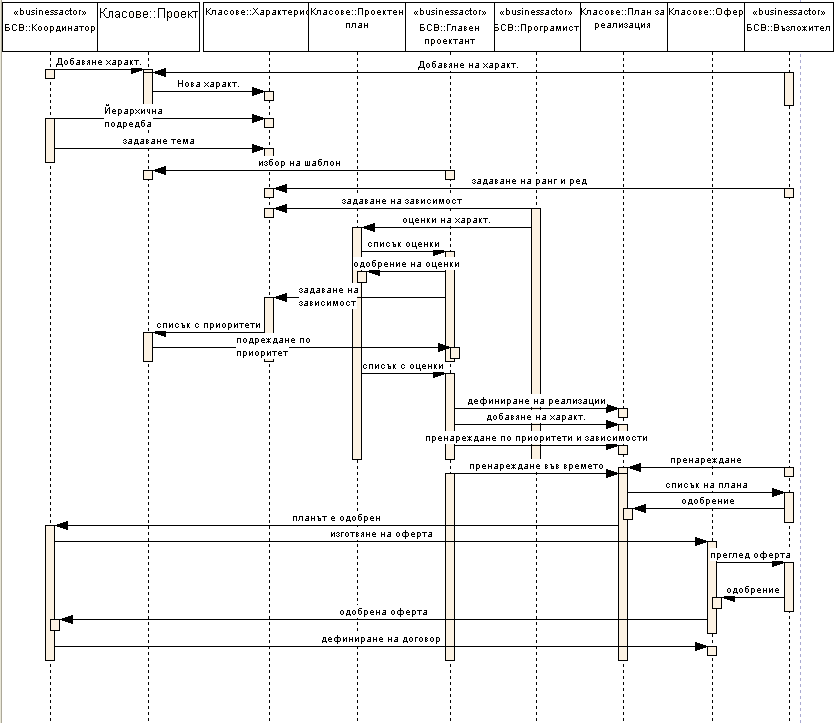
Фиг. 2.6. Диаграма на класовете и техните връзки за „Разработка на продукт”. (разработка на автора)



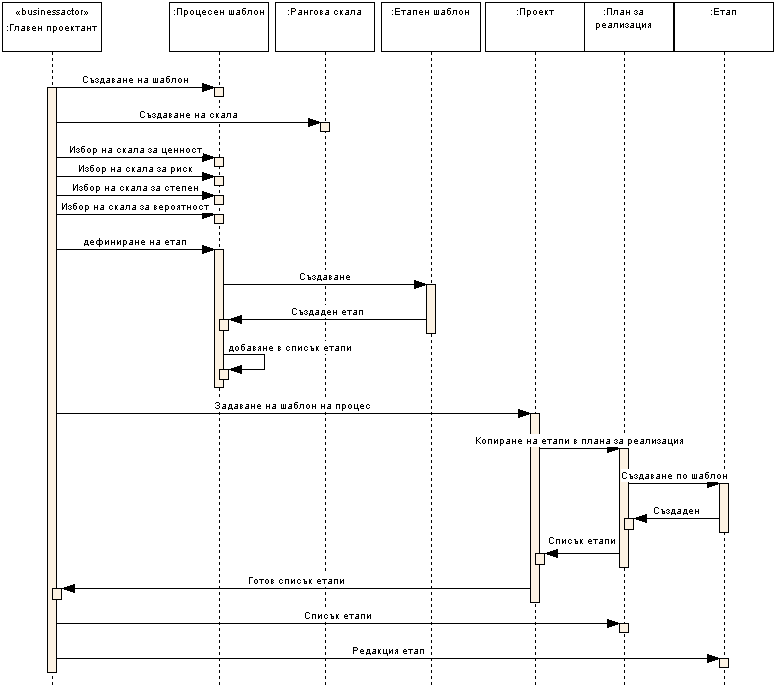
Фиг. 2.7. Диаграма на съществените класове за шаблони, формирани от бизнес същностите. (разработка на автора)

### 2.2.2. Диаграма на бизнес сценарии за взаимодействие със системата

По отношение на динамичните аспекти са разработени диаграми на дейности, реализиращи сценариите, където се показва начинът, по който протича процесът чрез дейности (фиг. 2.2). Тези дейности се детайлизират в диаграми на последователност и комуникационни диаграми (колаборативни, на сътрудничество) за класовете, за да се покаже начинът, по който дейностите се реализират чрез взаимодействие между класовете във времето и се осъществява съвместната им работа. За пример е дадена диаграма на последователност на фиг. 2.8 по отношение на дейността „планиране на проект” на база диаграмата за дейност на фиг. 2.2, т.е. това представлява диаграма за поведение на класове, с които се реализира дейността по БСВС „Планиране на проект”. Друга диаграма с голямо значение е дадена на фиг. 2.9 за последователност на реализацията на механизма за шаблони за процес, за рангови скали за ценност, риск, вероятност, степен и други, за да се реализира специфичен процес и подреждане. Фигурата е реализирана чрез диаграма за последователност на действията.



Фиг. 2.8. Интерактивна диаграма за БСВС „Разработка на продукт / Планиране на проект”. (разработка на автора)



Фиг. 2.9. Диаграма за реализация на шаблони за процес. (разработка на автора)

На тази база може да се определят операциите на отделните класове, което спомага както да се завърши дефиницията за интерфейса на бизнес логиката, така и функционалността на системата като необходим интерфейс. Допълването с диаграми на състоянията за статуса на класовете, които бяха представени в информационния модел на концепцията (фиг. 2.4), също е от съществено значение. Те посочват жизнения цикъл на обектите и необходимата логика при управлението им съгласно техния статус и събития.

### 2.2.3. Разположение на компонентите на системата по слоеве и насоки за изработка на работни продукти

По отношение на **компонентите на системата**, за да се реализира софтуерната система съгласно избраната технология, тя е структурирана в отделни логически компоненти. Компонентите са модули, които осигуряват определена функционалност като част от софтуерната система. Тези части са разположени на отделните логически слоеве в използваната тук трислойна архитектура. Технологично, на физическо ниво, тези компоненти се разполагат на съответен хардуер или виртуална машина – сървър за база данни, приложен сървър за бизнес логика и уеб достъп, и клиентски машини. При адаптацията на предложения подход се осигурява разделяне на слоевете и функциите с възможност за капсулиране, по-добра защита и поддръжка, и оптимизация на натоварването. По изключение може да се съвмести сървър за база данни и приложен сървър на една машина, ако е защитена от мрежови атаки. Описание на компонентите, разпределени по слоеве и хардуер, е дадено в табл. 2.1.

Таблица 2.1.  
 Разположение на компоненти по слоеве и хардуер  
(разработка на автора)

| **Слой** | **Хардуер/ВМ** | **Компонент** | **Пояснения** |
| --- | --- | --- | --- |
| **Слой за данни** | Сървър за БД | База данни на софтуерната система | За съхранение на данни относно продукти, проекти, процеси , потребители и други от софтуерната система. Реализира се с релационна БД. |
|  | Хранилище за документи | За съхранение на документация. Може да е портал на проекта, от рода на MS SharePoint, система за управление на конфигурации или прост файлов сървър. |
| Приложен сървър | Компонент за достъп до данни | Контролира и подпомага достъпа до данни в БД от бизнес логиката. |
| **Слой за бизнес логика** | Приложен сървър | Компонент за бизнес логика | Предоставя функции на логическо ниво и спазване на правила в софтуерната система. Контролира достъпа до данни. |
|  | Компонент за автоматизация на процеси | Подпомага автоматизацията, прилага правила за промени и вземане на решения, както и правила за уведомление. |
|  | Компонент за потребителски достъп | Потребители, оторизация, назначения, права за достъп. |
|  | Компонент за уведомления | За уведомяване на потребителите. Генериране лично/ автоматично от софтуерната система, изпращане под различна форма – лог, email, други. |
|  | Компонент за генериране на отчети | Генериране на данните за отчети и секции от лични табла. Генериране на отчетите с настройки. |
| **Представителен слой** | Приложен сървър | Уеб сървър | Сървър за хостване на сайт и уеб услуги. |
|  | Портал за работа на потребители | Сайт за достъп от потребители. |
|  | Компонент за програмен достъп | За програмен достъп към уеб услуги. |
|  | Компоненти за потребителски интерфейс | Компоненти за интерфейс – генериране на графики, изобразяване на отчети, AJAX и други интерфейсни. |
| Клиентска машина | Браузер за връзка с уеб приложение | Браузер за връзка с уеб сървъра и уеб приложението на софтуерната система. |

Предлаганият от нас подход изисква изработка на **работни продукти (РП)** ипри технологична стартираща компания те може да са под формата на документи, но част от тях са вградени в предлаганата софтуерна система. Някои са заимствани от фундаментални, според нас, разработки (Cockburn, 2021), други са преработени или добавени от нас на база личния опит по работа върху различни софтуерни проекти. Списъкът на работните продукти е следния:

– Визия – съдържа една-две страници описание на проект и цели, и списък на приоритети (критични, възможни или жертвани) за характеристики на софтуера[[9]](#footnote-10);

– Екипна структура – обозначава начините, по които отделните членове на екипа комуникират – вградено е в системата с потребители, роли и назначения на потребители към проект с дадени роли;

– Конвенции за работа – списък от правила, практики и обичаи, които екипът спазва в хода на съвместната работа[[10]](#footnote-11);

– Списък сценарии – списък от тип „Actor-Goal” за сценарии с отговорници, вграден в софтуерната система;

– Сценарии за взаимодействие – представен е във форма на модел или текстово описание – конкретно за тази софтуерна система това са бизнес сценарии на взаимодействие от ИБМ, дадено в концепцията;

– Модел на потребителски роли – списък роли в изработвания софтуер, аналогия на роли в софтуера, бизнес актьори и работници в ИБМ;

– Списък характеристики – списък на характеристики с подреждане и зависимости в свободна форма като списък или мрежови график – вградени са като списък/дърво в софтуерната система;

– Изисквания – списък изисквания в свободен текст от потребителите – вграден като списък и описание на характеристики в софтуерната система[[11]](#footnote-12);

– Оферта – включва параметри за договор/план – срок, цена, условия, ограничения, списък с характеристики, обобщени оценки – добавена е от автора и вградена в софтуерната система;

– Договор – условия и параметри на договора с приложен план – добавен е от автора;

– План за реализации – срокове и разпределени характеристики – вграден в софтуерната система;

– Проектен статус – списък или диаграма с реализации, статуса им и бележка какво е направено; вградено в софтуерната система, но може да се подготви като „информационно табло” с маркер за статус;

– Списък рискове – съдържа списък на рискове по приоритет; описва се за вероятност, отговор, промяна на приоритет и засегнати итерации – вградено в софтуерната система като „информационно табло”;

– План/статус на итерация – списък на задачи и статуса им в текуща итерация;

– План за прегледи – списък на предстоящи реализации за уведомяване на екип и потребители; вградено в софтуерната система като списък реализации – може да е „информационно табло”;

– Програмистки продукти – продукти от проектант/програмист, които са за проектиране/реализиране на софтуера[[12]](#footnote-13); Вид, форма, детайли и съхранение на документите се избират от екипа (или описани в конвенциите); Софтуерната система поддържа само връзки към файлове;

– Тестове – списък автоматизирани тестове или описание за ръчни тестове за код и за приемане на софтуерния продукт от възложител – софтуерната система поддържа само списък на групи или на приемни тестове, които да се извършат (с описание или връзка към документ);

– Отчет на дефекти – списък с получени грешки при работа на софтуера в резултат на тест или забелязан проблем – софтуерната система ги поддържа като проблеми, а програмист конвертира проблемите в дефекти и ги поправя, като за софтуерната система са от тип задачи с работа за вършене;

– Помощен текст – от редактора помощен текст за ползване в системата;

– Потребителска помощ – структурирано описание от редактор на софтуера[[13]](#footnote-14);

– Ръководство за потребителя – упътване от редактора за обучение на потребители[[14]](#footnote-15);

– Резултати от отразяващо подобрение – обичайно списък в три групи за това, което „Работи добре”, „Ще опитаме да променим” и „Нерешени проблеми” – вградено е в софтуерната система.

Някои РП могат да са информационни табла за обща/постоянна видимост, а други могат да са вградени в софтуерната система като документи, съхранявани в хранилището за документи.

## 2.3. Функционалност и потребителски интерфейс

По отношение на **функционалност и потребителски интерфейс**, представени в предходната глава е дефиниран в детайли интерфейса на потребителското приложение. Примерна скица на потребителския интерфейс е даден на фиг. 2.10 със следните основни елементи:

– Заглавна част – лого, заглавие, текущ потребител и дата, връзка за помощ;

– Главно меню – статично меню с опции видими според правата на потребителя;

– Контекстно подменю според избраното в главното меню и ролята;

– Маркер за наличие на нови съобщения, както и списък на спешни необработени такива;

– Списък на уместни и свързани проблеми, рискове, мерки за подобрение относно проекта и други данни свързани с текущата функция, роля или потребител;

– Списък на уместни/свързани съобщения относно текущия екран;

– Работна площ – място за работа на потребителя с екрани от опции на подменю;

– Завършваща лента – данни за продукт, версия, доставчик, връзки за помощ, обратна връзка за поддръжка и контакт с доставчик на софтуерната система.



Фиг. 2.10. Скица на основен екран на приложението. (разработка на автора)

Потребителят има достъп до заглавната част, главното меню и завършващата лента винаги. В повечето случаи е видимо контекстното подменю, списъци с новини и свързани неща. След вход в системата с оторизация, се показва началният екран. Той също притежава гореизброените елементи, но на мястото за работна площ има лично (персонализирано) табло с кратки бързи справки за статус, цветна (светофарна) сигнализация, евентуално предстоящи срокове, кратки списъци с връзки за текущи проекти, задачи, проблеми и други най-важни елементи. Тези елементи обичайно са кратки, като данните са подредени по приоритет или дата и ограничени като брой (например 5-те най-важни, 10-те най-закъснели).

Списъците имат бързи връзки към детайлен екран за съответния обект и функция. Личното табло дава само най-важната информация за проект/продукт, статуса му и процеса. При избор на опция от главното меню се появява подменю със списък опции, а на работната площ може да се появи или работният екран на първа опция от подменюто (обичайно преглед на данни) или вид лично табло специфично за избраната опция в главното меню. потребителя. Главното меню и подменютата имат видими опции, според правата за достъп на ролите на потребителя. Личното табло има секции, съгласно ролите на потребителя, а данните в него може да се филтрират за конкретния потребител.

В екраните за отделни функции, разположени в работната площ, е налице обичайна последователност. Най-отгоре са разположени секции и бързи връзки към други екрани в специална лента за страници/връзки. Под нея се разполага обща информативна секция за избран продукт, проект, итерация или друго, и/или евентуален филтър за бърз преход (ако е необходим). Под тях е самият екран за преглед/редакция, като в долния му край обичайно са бутоните за действия. За всички списъци има бутони/ връзки в дясната страна за стандартни бързи операции като избор, преглед, обновяване, приключване и т.н.

Списъците се подреждат по предназначението си или с клик на етикетите на колоните. Списъците по принцип нямат страниране, освен ако не са дълги (над 50 реда). Софтуерната система и екраните й предвиждаме да имат възможност да запомнят последен избор на потребителя и да предоставят филтър за това (например последен избран проект). В друг екран, който изисква избор и филтър, то той е избран автоматично за последното. Така преминаването между функции със запазен контекст улеснява потребителя, интерфейса и създава фокус за работа.

Относно отчетите, е удачно да се приложи гъвкава (лесно адаптируема) система, позволяваща както извикване на отчет от специален екран за заявка на отчет, така и директно от връзка в друг екран с параметри по подразбиране. В екрана за заявка на отчет се избира групата и вида отчет, задават се параметри като продукт, проект, реализация, итерация, служител, срок и евентуално друг частен критерий за даден отчет. Възможен е критерий за групиране на отчетите по време – дневни, седмични, месечни, тримесечни. Отчетът може да се заяви на екран или данните му да са изведени в CSV формат. При извеждането на екран той показва заглавие, избрани параметри и групиране, таблица с данните и секция за графика.

Основните функции и менюта с кратък коментар за тях са представени в табл. 2.2 – главно меню, подчинени менюта, както и свързани екрани.

Таблица 2.2.  
 Структура на менюта и потребителски интерфейс в клиентското приложение.  
(разработка на автора)

| **Главно/ Подчинено меню** | **Описание** | **Връзки с други екрани** |
| --- | --- | --- |
| **Начало** | Главно лично табло със секции за сигнализация, списък и статус на достъпни проекти, графика за очакван край и степен изпълнение, други най-важни области. |  |
| Проекти | Преминава в Проекти / Списък на мои проекти |  |
| Задачи | Преминава в Итерации / Списък на мои задачи |  |
| Проблеми | Преминава в Проекти / Списък на проблеми |  |
| Дефекти | Преминава в Проекти / Списък на дефекти |  |
| Съобщения | Преглед на лога със съобщения касаещи потребителя. | Добавяне/изпращане на ново съобщение. |
| **Продукти** | Показва лично табло |  |
| Лични | Лично табло с текущи продукти и проекти | Списък продукти, нов продукт, Мои проекти, Добавяне на проект |
| Списък продукти | Азбучен списък с филтър – архивирани, търсене | Детайли за продукт, Нов продукт, Нова/ Редакция версия, Редакция клиент |
| Нов продукт | Може да се дефинира нов продукт. |  |
| **Клиенти** | Списък клиенти |  |
| Списък клиенти | Азбучен списък на клиенти с филтър – текущи проекти, архивирани, търсене | Редакция/ Добавяне на клиент, Списък/ Добавяне на представител, Списък продукти, Нов продукт, Списък проекти, Нов проект |
| Списък представители | Азбучен списък представители с филтър – текущи проекти, архивирани, търсене | Редакция на представител, Редакция на клиент, Добавяне на клиент |
| **Проекти** | Активни проекти |  |
| Активни | Азбучен списък на проекти | Списък проекти с филтър, Редакция проект, Добавяне на проект, Преглед реализация, Преглед итерация, Преглед / добавка на записки и документи; Отчети. |
| Списък проекти | Азбучен списък на проекти с филтър за име, код, срок, клиент, др. | Почти същите без списък; Отчети. |
| Добавяне на проект | Дефиниране на нов проект, условия и ограничения, избор на шаблон за процес и на рангови скали |  |
| Редакция проект | Филтър за проект, Редакция на данни, срок, условия | Процес, Редакция шаблони за процес, Редакция на рангови скали |
| Процес | Филтър за проект, Списък етапи, Редакция на процес с етапи и данни за тях, избор на рангови скали (ако статуса позволява промени) | Редакция шаблони за процес, Редакция на рангови скали, |
| Сценарии | Филтър за проект. Дефиниция на списък сценарии (работен продукт „Action Goal List”), преобразуване в характеристики. | Връзка с Характеристики |
| Характеристики | Филтър за проект и тема на характеристики. Йерархичен списък характеристики. В един екран обновяване, добавяне, изтриване, преподредба, смяна на статус (ако статуса на проекта позволява). | Преглед и добавка на записки и документи, |
| Подреждане на характеристики | Филтър за проект и тема на характеристики. Списъкът с характеристики се подрежда за стойност и риск по избрана скала. Задават се и зависимости от вида – X преди Y. | Редакция на рангови скали |
| Оценка на усилия | Филтър за проект и тема на характеристики. Оценка на характеристиките за сложност и усилия. |  |
| Договори | Филтър за проект. Списък договори, промяна на договор, добавяне на нов, задаване на параметри, смяна на статус, одобрение, приключване или отказ, справка за усилия според оценките. | Редакция договор, Нов договор, Оферта за проект |
| **Реализации** | Преглед реализация |  |
| Преглед | Филтър за проект и реализация с автоматичен избор на текущата. Показва кратка статистика – брой характеристики, завършени и време за оставащи, др. Кратък списък итерации и рискове, както и данни за реализацията и прогноза за очаквано завършване. | Редакция реализация, Характеристики в реализация, Планиране на реализация |
| Редакция | Филтър за проект. Списък на реализации. Редакция на избрана или добавяне на реализация, дефиниране на срок и други параметри. |  |
| Характеристики | Филтър за проект и за реализация (по подрзбиране текуща), за тема на характеристики. Йерархичен списък на всички и линеен списък на избрани характеристики. Възможност за преглед и ако статуса на реализацията позволява – промяна на избраните и статуса им. | Планиране |
| Планиране | Филтър за проект и автоматичен избор на текуща реализация. Филтър за тема на характеристики. Йерархичен списък на всички и линеен списък на избрани характеристики. Възможност за добавяне, изтриване и преоценка на избраните (ако статуса го позволява). |  |
| Списък рискове | Филтър за проект и реализация, автоматичен избор на текуща реализация. Списък рискове за реализацията подреден по ранг. Добавяне, редакция и премахване от списъка, смяна на ранг. | Назначаване на задача за риск |
| Подобрение | Филтър за проект и реализация, автоматичен избор на текуща реализация. Списък мерки от подобрението по групи и възможност за редакция, добавяне, махане, назначаване на задачи от тях. | Назначаване на задача за подобрение |
| **Итерации** | Преглед (с формата на лично табло) |  |
| Преглед | Филтър за проект, реализация и итерация с автоматичен избор на текущи. Статистика за задачи и дефекти (брой, завършени, оставащо време), проблеми, тестове, рискове, оставащо време и др. | Редакция итерация, Задачи в итерация, Дефекти в итерация, Планиране на итерация |
| Редакция | Филтър за проект, реализация и итерация с автоматичен избор на текущи. редакция на итерация, срок и т.н. |  |
| Планиране | Филтър за проект, текуща реализация и итерация избрани автоматично. Списък задачи, тестове и дефекти. Възможност за пренареждане, оценка на задачите. Справка общо и по потребител за съотношение капацитет време/назначени задачи коригирани с параметрите за индивидуална преценка. | Добавяне на задача, на тест или дефект. Добавяне на задача за риск. |
| Подобрение | Филтър за проект и реализация, автоматичен избор на текуща реализация. Списък мерки от подобрението по групи и възможност за редакция, добавяне, махане, назначаване на задачи от тях. | Назначаване на задача за подобрение |
| **Работа** | Преглед (с формата на лично табло за текуща работа) |  |
| Преглед | Филтър за проект, реализация и итерация са автоматично избрани текущите. Показва сигнализация за статус на итерация, статистика за задачи (брой и завършени), дефекти (общо и завършени), списък на задачи и дефекти подредени по приоритет на характеристики или по дата на завършване, списък проблеми, рискове, и други. | Преглед, добавяне и редакция на задача, преоценка на задача, добавяне на дефект, добавяне на проблем |
| Задачи | Филтър за проект, реализация и итерация са автоматично избрани текущите. Списък задачи по ред или приоритет. Бързи връзки за смяна на статус, преоценка, оставащо време. |  |
| Дефекти | Сходно на задачи. Бърза добавка, смяна статус, преоценка, време. |  |
| Проблеми | Сходно на задачи. Бърза добавка, смяна статус и конверсия. |  |
| **Отчети** | към Филтър на отчети |  |
| Преглед | Филтър за избор на група отчети и конкретен отчет. Филтър за продукт, проект, реализация, итерация с автоматичен избор на текущите. Филтър за клиент, период. Избор групиране. | Генератор на отчет – генератор, който според избрания отчет и параметри го генерира като таблици с данни, графики. |
| **Администрация** | Празен екран. |  |
| Шаблони | Редакция и добавяне на шаблони. | Нов шаблон. |
| Рангови скали | Редакция и добавяне на на рангови скали. Защита при употреба. | Нова скала. |
| Списъци | Редакция на списъци използвани в системата с възможна промяна. | Нов списък. |
| **Профил** | Преглед и редакция на профил на потребителя |  |
| Редакция профил | Преглед на данните и лично обновяване на профила на абоната |  |
| Смяна на парола | Смяна на паролата с потвърждение на старата |  |

Таблицата е възможно да се допълва и разширява, т.е. тя не е изчерпателна и не детайлизира всички функции, но според нас дава съществена представа за главните екрани и посочва допълнителните.

***Изводи и обобщения към втора глава***

1. Разработен е концептуален модел на софтуерна система, управляваща основните процеси и дейности при разработката, поддръжката и съпровождането на софтуер.

За малките компании са характерни по-ниската степен на формализация, опростени процедури и правила. Затова по отношение на обхват на софтуерната система, приемаме че разработка на софтуер покрива определен набор от функционалности, като се избягва голямата сложност и влизане в чисто технически въпроси относно проектиране, програмиране, тестване и интеграция, управление на документи, конфигурации и т.н. За тях може да се ползва специализиран софтуер по индивидуална преценка.

При разработката на модела сме се ограничили единствено над основния бизнес сценарии за взаимодействие със системата, за който са разработени диаграми на взаимодействията и колаборативни диаграми, с които да се показва начина на взаимодействие между служителите с различни роли в технологичната стартираща компания и обхвата на техните отговорности. Техните отговорности са основа за формиране на функциите на системата и персонализираният достъп за всяка роля.

2. Разработен е логически модел на софтуерна система.

Логическият модел е разработен на базата на предложения концептуален бизнес модел и за реализацията му в началото са обособени и развити бизнес същностите като класове със съответни свойства и връзки.

# Използвана литература

1. Галоуей, С. (2021) *Посткорона. От криза към възможност*. Изток-Запад, София.
2. Ескенази, А., & Манева, Н. (2006). *Софтуерни технологии.* София: КЛМН.
3. Закон за защита на конкуренцията. Държавен вестник, София, бр.17 от 26.02.2021 г.
4. Закон за корпоративното подоходно облагане. Държавен вестник, София, бр. 95 от 8.12.2015 г.
5. Закон за малки и средни предприятия. Държавен вестник, София, бр. 17 от 1.03.2016 г.
6. Закон за насърчаване на инвестициите. Държавен вестник, София, бр. 61 от 11.08.2015 г.
7. Закон за счетоводството. Държавен вестник, София, бр.19 от 05.03.2021 г.
8. Илиев, П., Сълов, В., & Петров, П. (2010). *Виртуални системи.* Монографична библиотека „Цани Калянджиев”, Варна: Наука и икономика.
9. Сълов, В. (2014). *Производителност и ефективност на компютърните системи*. Варна : Унив. изд. Наука и икономика.

**Интернет източници**

1. Даскал, Л. (2018). 7 стратегически умения за по-ефективно лидерство. Мениджър. <https://manager.bg/liderstvo/trayna-7-nachina-da-bdete-po-efektivni-lideri> [23.08.2022]
2. Икономи, П. (2018). Тайните на ефективното лидерство. Мениджър. <https://manager.bg/liderstvo/trayna-taynite-na-efektivnoto-liderstvo> [23.08.2022]

1. UML има възможности за детайлно описание на взаимодействие с външни партньори и системи, както и за вътрешна архитектура и поведение на ИС. За целите на ИБМ използваме предимството на UML за разширение чрез специализация на същия метод и елементи в желаната област на моделиране. [↑](#footnote-ref-2)
2. Има се предвид само екип от компанията – главен проектант, програмисти, тестери, редактори, координатор; [↑](#footnote-ref-3)
3. При създаване на моделите в труда е използван софтуер Enterprise Architect на компанията Sparx Systems (https://sparxsystems.com/). Ползваната анотация от графични елементи е съгласно спецификацията за UML 2.5. [↑](#footnote-ref-4)
4. Това е структуриран (XML, named pairs) списък от нестандартни данни с име и стойност. Потребителят/ИС ги добавя за/от интеграция с външен продукт или допълнителни записки. [↑](#footnote-ref-5)
5. За рангова скала за ценност на характеристика е подходящо да се използва например HML (high, middle, low), MoSCoW от DSDM, и др.; [↑](#footnote-ref-6)
6. Например, на база лични наблюдения и опит, можем да твърдим, че характеристиките в малък проект са обичайно 3-4 групи по 10-12 с няколко детайла. Общо до 50-100 на брой. Структурирането и разделението по реализации намалява тази бройка. [↑](#footnote-ref-7)
7. Има се в предвид работа с един-два клика на мишката, един-два бързи клавиша и съкратено въвеждане на допълнителни данни. [↑](#footnote-ref-8)
8. Според нас, в малък екип е излишна сложност да се отделят време и ресурси с твърде детайлно планиране на ниво итерация. Включено е планиране на зависимости между характеристиките на ниво проект, което трябва да се съобрази при разпределението на характеристиките в реализациите. В по-голям проект и екип работата може да се преструктурира, така че да се намалят проблемите със зависимостите. [↑](#footnote-ref-9)
9. Предлагаме да се разглеждат като вид насока в какво да се фокусира екипа – скорост, функции и т.н., като формират визията на софтуерната система. [↑](#footnote-ref-10)
10. Посочва се избрания подход за разработка, както и указания за възлагане на роли, за кодиране, собственост на код, преглед на код, начин на работа и отчитане в итерации, преглед от потребител, управление на конфигурации и др. [↑](#footnote-ref-11)
11. По наше мнение е подходящо и препоръчително в компанията да има приети общи шаблони за еднотипен софтуер, които ориентират клиента с въпроси. [↑](#footnote-ref-12)
12. Включват модел на областта (например обектно ориентиран модел с UML), системна архитектура, дизайн на екрани, други програмистки скици и модели, сорс код, инсталационен пакет, дизайн, бележки. [↑](#footnote-ref-13)
13. Потребителската помощ може да има стандартен формат на структура за всички точки, например: йерархия в глави, раздели, точки с цел, описание, важно, изисквания, връзки; може да се автоматизира с експорт на реализирани характеристики. [↑](#footnote-ref-14)
14. Разликата с потребителската помощ е, че не е структурирана помощ по теми, а е във вид на учебник или ръководство. [↑](#footnote-ref-15)