



Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика
Катедра „Софтуерни технологии“

Автореферат
на дисертационен труд на тема
Управление на знания в софтуерните процеси

Докторант: Дилян Иванов Георгиев
професионално направление 4.6 „Информатика и компютърни науки“
докторска програма „Софтуерни технологии“ – Управление на знания

Научен ръководител:
доц. д-р Елисавета Гурова
Катедра „Софтуерни технологии“
ФМИ, СУ „Св Климент Охридски“

София 2025 г.

Съдържание

| | |
|---|----|
| Терминологичен речник и използвани съкращения | 4 |
| Списък с фигурите | 7 |
| Списък с таблиците..... | 7 |
| 1. Увод..... | 8 |
| 1.1. Мотивация за избор на темата..... | 8 |
| 1.2. Обект и предмет на изследването | 9 |
| 1.3. Цели и задачи на дисертационния труд | 9 |
| 1.4. Приложени методи..... | 10 |
| 1.5. Структура и съдържание на дисертационния труд | 10 |
| 2. Проучване на текущото състояние | 12 |
| 2.1. Управление на знания и документи | 12 |
| 2.2. Специфика на УЗ в софтуерното инженерство | 12 |
| 2.3. Области на знания в софтуерните процеси | 13 |
| 2.4. Технологии, подпомагащи УЗ в СИ | 14 |
| 2.4.1. Хранилища на документация | 14 |
| 2.4.2. Техники за документация | 14 |
| 2.4.3. Колаборативни платформи и инструменти | 14 |
| 2.4.4. Системи за обучение и обмен на знания | 15 |
| 2.4.5. Системи за управление на документация..... | 15 |
| 2.5. Сравнителен анализ на системите за управление на документация | 16 |
| 2.6. Нови тенденции в управлението на знания и документи..... | 16 |
| 2.7. Изводи..... | 17 |
| 3. Проучване на управлението на знания в софтуерните компании..... | 18 |
| 3.1. Систематизиране на областите на знания в софтуерното инженерство | 18 |
| 3.2. Знания за контекста в Софтуерното инженерство..... | 19 |
| 3.2.1. Знания за продукта | 19 |
| 3.2.2. Знания за процеса..... | 20 |
| 3.3. Организационни знания в софтуерното инженерство | 20 |
| 3.3.1. Технократски знания..... | 20 |
| 3.3.2. Поведенчески знания | 21 |
| 3.4. Проучване на потребностите от управление на знания в софтуерните процеси . | 21 |
| 3.4.1. Методология на проучването..... | 21 |

| | |
|---|----|
| 3.4.2. Анализ на резултатите | 22 |
| 3.5. Изводи..... | 26 |
| 4. Приложение на изкуствен интелект при управление на знания в софтуерните процеси..... | 28 |
| 4.1. Концептуална рамка за Автоматизирана система за управление на документи | 28 |
| 4.2. Проучване за основните функционалности в АСУД | 29 |
| 4.2.1. Методология на проучването..... | 29 |
| 4.2.2. Анализ на резултатите | 29 |
| 4.3. Проектиране на АСУД | 30 |
| 4.3.1. Основни слоеве и компоненти на системата | 31 |
| 4.3.2. Управление на достъпа и потребителите | 32 |
| 4.3.3. Интеграция с ИИ..... | 32 |
| 4.4. Разработване на прототип на АСУД | 33 |
| 4.4.1. Функционални изисквания | 33 |
| 4.4.2. Нефункционални изисквания и ограничения | 34 |
| 4.4.3. Дизайн и софтуерна архитектура на прототип на АСУД | 34 |
| 4.4.4. Реализиране на прототипа | 36 |
| 4.4.5. Внедряване на прототипа в реална среда | 38 |
| 4.5. Изводи..... | 38 |
| 5. Валидация на прототип на АСУД..... | 40 |
| 5.1. Методология на валидацията | 40 |
| 5.1.1. Изготвяне на тестови сценарии | 40 |
| 5.1.2. Подготовка на експеримент | 40 |
| 5.2. Анализ на резултатите | 40 |
| 5.3. Изводи..... | 42 |
| 6. Заключение..... | 44 |
| 6.1. Ползи от предложената АСУД | 44 |
| 6.2. Постигнати резултати и бъдещи възможности..... | 44 |
| Основни приноси на дисертационния труд | 46 |
| Научно-приложни приноси | 46 |
| Приложни приноси..... | 46 |
| Публикации по дисертацията | 46 |
| Референции | 47 |

Терминологичен речник и използвани съкращения

| Абревиатура | Английски термин | Български еквивалент |
|-------------|---|--|
| API | Application programming interface | Интерфейс за приложно програмиране |
| CAMS | Culture, Automation, Measurement, Sharing | Култура, Автоматизация, Измерване, Споделяне |
| CASE | Computer-Aided Software Engineering | |
| DevOps | Development and Operations | Разработка и операции |
| EDM | Electronic document management | Електронно управление на документи |
| GenAI | Gen AI | Генеративнен ИИ |
| JSON | JavaScript Object Notation | Нотация за описание на обекти в JavaScript |
| LLMs | Large Language Models | Големи езикови модели |
| PR | Pull Request | Заявка за сливане на код |
| QA | Quality Assurance | Осигуряване на качество |
| QDAD | Quality-driven Architecture Design | Дизайн на архитектура, управляван от качеството |
| Q-Rapids | Quality-Aware Rapid Software Development Design | Качествено-осъзнат гъвкав дизайн за софтуерна разработка |
| RE | Requirements engineering | Инженеринг на изискванията |
| SDLC | Software development lifecycle | Жизнен цикъл на софтуерната разработка |
| UML | Unified Modelling Language | Унифициран език за моделиране |
| UX/UI | UX/UI | Потребителско изживяване / Интерфейс |
| VCS | Version Control System | Система за контрол на версиите |
| VEL | Variability Exchange Language | Език за обмен на променливост |

| | | |
|------|--|--|
| WADL | Web Application Description Language | Език за описание на уеб приложения |
| АСУД | Automated Documentation Management System (ADMS) | Автоматизирана система за управление на документация |
| ЕСУД | Electronic Document Management System | Електронна система за управление на документи |
| ИИ | Artificial Intelligence (AI) | Изкуствен интелект |
| ИКТ | Information and Communication Technologies | Информационни и комуникационни технологии |
| ИС | Information Systems | Информационни системи |
| ИТ | Information Technology | Информационни технологии |
| РНМ | Recurrent Neural Networks | Рекурентни Невронни Мрежи |
| СИ | Software Engineering | Софтуерно инженерство |
| СУД | Document Management System | Система за управление на документи |
| СУЗ | Knowledge Management System (KMS) | Система за управление на знания |
| УЗ | Knowledge Management | Управление на знания |
| ЧР | Human Resources | човешки ресурси |
| РНМ | Recurrent Neural Networks | Рекурентни Невронни Мрежи |
| - | Agile Methodologies | Гъвкави методологии |
| - | Bugs | Бъгове (грешки в софтуера) |
| - | Business analyst | Бизнес анализатор |
| - | Chatbot | Чатбот / чат асистент |
| - | Gamification | Геймификация |
| - | Gantt chart | Гант диаграма |
| - | Intelligent assistant | Интелигентен асистент |
| - | Iterative model | Итеративен модел |
| - | Knowledge domain | Област на знания |
| - | Logs | Логове |

| | | |
|---|--------------------|-----------------------------|
| - | Merge conflict | Конфликт при сливане на код |
| - | Project manager | Мениджър на проекти |
| - | Prototyping | Прототипиране |
| - | Quality specialist | Специалист по качеството |
| - | Screenshots | Екранни снимки |
| - | Source Code | Изходен код |
| - | Spiral model | Спираловиден модел |
| - | Technical debt | Технически дълг |
| - | Test scenario | Тестови сценарий |
| - | User-friendly | Удобен за потребителя |
| - | V-shaped model | V-образен модел |
| - | Waterfall model | Водопаден модел |

Списък с фигурите

| Номер | Име | Референция |
|------------|--|------------|
| Фигура 3.1 | Визуална схема на основни концепции в СИ и УЗ идентифицирани в данните на изследването | 3.1 |
| Фигура 3.2 | Документи в жизнения цикъл на софтуерната разработка | 3.2.1 |
| Фигура 3.3 | Оценка на знанията за продукта според опита и големината на компанията | 3.4.2 |
| Фигура 3.4 | Оценка на знанията за процеса според опита и големината на компанията | 3.4.2 |
| Фигура 3.5 | Оценка на технократските знания според опита и големината на компанията | 3.4.2 |
| Фигура 3.6 | Оценка на поведенчески знания според опита и големината на компанията | 3.4.2 |
| Фигура 4.1 | Мястото на АСУД спрямо различните области на знания | 4.1 |
| Фигура 4.2 | Концептуална рамка за Автоматизирана система за управление на документи | 4.1 |
| Фигура 4.3 | Архитектурен модел на АСУД | 4.3 |
| Фигура 4.4 | Изглед на администратор | 4.4.4.2 |
| Фигура 4.5 | Изглед на обикновен потребител | 4.4.4.2 |
| Фигура 4.6 | Изглед на файлово дърво за управление на документи | 4.4.4.2 |
| Фигура 4.7 | Изглед на компонентата за интелигентен асистент | 4.4.4.2 |

Списък с таблиците

| Номер | Име | Референция |
|--------------|--|------------|
| Таблица 3.1. | Оценка на знанията за продукта според опита и големината на компанията (t-тестове) | 3.4.2 |
| Таблица 2.1 | Сравнителен анализ на системи за управление на документи | 2.5 |
| Таблица 4.3 | Основни функционалности на прототипа | 4.4.1 |
| Таблица 5.1 | Статистическа оценка на прототипа (t-тестове) | 5.2 |

1. Увод

В съвременния свят, белязан от бързи технологични промени и нарастваща конкуренция, ефективното управление на знания (УЗ) се превръща в основен фактор за успеха на организациите. Най-често УЗ се определя като систематичен процес на събиране, организиране, съхраняване, споделяне и използване на информация и експертен опит с цел повишаване на ефективността, подобряване на взимането на решения и ускоряване на иновациите (Gourova, 2012).

Технологичните компании и изследователските институции използват системи за управление на знания (СУЗ), за да оптимизират процесите си чрез централизиране на информацията, улесняване на сътрудничеството и автоматизация на задачи (Polyakov et al., 2020). В същото време софтуерната разработка е изключително специализирана дейност, при която всеки служител може да развие уникални знания и експертиза, които трудно могат да бъдат събрани и документирани от екипите или компаниите. Тази експертиза е свързана с работата в различни информационни системи (ИС), които правят "ясно разграничение между данни, информация и знания" (Baidoun and Ahmad, 2021) и прилагат подходи за персонализиране.

Ролята на ИИ в улесняването на създаването на знания и иновациите е критична за цялостната еволюция на знанията (Jarrahi et al., 2023). Чрез усъвършенствани анализи и техники за разпознаване на модели с ИИ може да се разкрият скрити зависимости и връзки в данните, което води до нови прозрения и открития, както и до оптимизиране на организационните стратегии (Borges et al., 2021). СУЗ с ИИ могат да помогнат на организациите да идентифицират и адресират възникващи предизвикателства, което повишава тяхната устойчивост и адаптивност в динамичната бизнес среда.

1.1. Мотивация за избор на темата

Темата на дисертационния труд е свързана с проучване на ключовите области на знания, които подпомагат софтуерните процеси и улесняват достъпа до документирани знания, свързани с разработването на софтуер. Специално внимание се обръща на ролята на изкуствения интелект в управлението на знания.

Необходимостта от това проучване произтича от неизследваната територия между софтуерното инженерство, движено от технологиите, и управлението на знания като механизъм за оптимизация. Това очертава ново изследователско поле, фокусирано върху методите и подходите, които могат да повишат ефективността и продуктивността на софтуерните процеси.

Разработването на автоматизирана система за управление на документи (АСУД) предлага съвременен решение на представените по-горе проблеми. Такава система има за цел да интегрира принципите на УЗ с практиките на софтуерното инженерство, като осигури централизирано, интелигентно и ефективно управление на документацията и натрупаните знания в процеса на разработка на софтуер.

Една от основните движещи сили зад тази система е необходимостта от по-добра структурираност и достъпност на знанията в софтуерните екипи. Често разработчиците, проектантите и ръководителите разчитат на разпокъсани източници на информация – лични бележки, разпръсната документация, неформални комуникационни канали. Това

не само затруднява работата, но и ограничава възможностите за учене и оптимизация на процесите, както и проследяване развитието на идеите.

С използването на изкуствен интелект и интелигентни агенти една автоматизирана система може не само да съхранява и организира информацията, но и да я анализира, обобщава и да прави препоръки в зависимост от конкретния контекст. Това би намалило времето за търсене на информация и би улеснило взимането на информирани решения, особено в критични фази на софтуерната разработка.

Всичко това показва, че реалните предизвикателства, свързани с разработването на софтуерни продукти, могат да бъдат значително подобрени чрез интегриране на интелигентни механизми за съхранение, анализ и достъп до информация. Такава система ще осигури по-добра ефективност, намаляване на рисковете и повишаване на качеството в софтуерната индустрия.

1.2. Обект и предмет на изследването

Обект на дисертационния труд е управлението на знания в софтуерните компании. Основният фокус е върху специфичните процеси в софтуерното инженерство и необходимостта от своевременно осигуряване на потребностите от знания за тяхното изпълнение чрез прилагане на съвременни технологии.

Предмет на изследването са системи за управление на документи, както и различните подходи за тяхното ползване в контекста на документирането на знания в софтуерното инженерство. От особен интерес са интелигентните системи, базирани на генеративен изкуствен интелект, и тяхната интеграция в системи за управление на документи с цел автоматизация, оптимизация на работните процеси и повишаване на ефективността на управлението на знания.

1.3. Цели и задачи на дисертационния труд

Настоящият дисертационен труд има за цел да проучи спецификата на управлението на знания в софтуерните процеси и да предложи АСУД, която да интегрира инструменти, базирани на изкуствен интелект.

Така поставената цел се реализира чрез следните задачи:

1. Да се проучи текущото състояние на управлението на знания в софтуерното инженерство, като се анализират използваните технологии за управление на знания и документи, както и идентифицират потребностите от управление на знания.
2. Да се идентифицират и систематизират областите на знания в софтуерното инженерство.
3. Да се разработи модел на областите на знания в софтуерното инженерство.
4. Да се проучат проблемите, свързани със знанията и дейностите в софтуерните процеси.
5. Да се разработи концептуална рамка на система за управление на документи в идентифицираните области на знания в софтуерното инженерство.
6. Да се предложи интеграцията на интелигентни агенти в системата за управление на документи.

7. Да се проектира и разработи прототип на система за управление на документи.
8. Да се валидира прототипа от експерти в областта на софтуерното инженерство.

Основната хипотеза на научното изследване предполага, че управлението на знания е важен компонент от софтуерния процес, който може да бъде значително подпомогнат чрез интеграцията на системи, използващи инструменти, базирани на изкуствен интелект. Тази хипотеза се основава на следните твърдения:

- Управлението на знания подпомага организациите да работят по-ефективно като използват наличната експертиза и знания в работните си процеси и при взимането на решения.
- Въпреки разнообразието от информационни системи, използвани в софтуерните процеси, софтуерните специалисти изпитват затруднения при намирането на необходимата им информация и знания.
- Интеграцията на ИИ инструменти в ИТ системите има потенциал да оптимизира работната среда, като подобрява достъпа и споделяне на знания и взимането на информирани решения.

1.4. Приложени методи

В рамките на изследването са приложени разнообразни научно-изследователски методи. Извършени са документални проучвания, сравнителен и критичен анализ на литературни източници. Използвани са конструктивни изследователски подходи и емпирични проучвания, включващи анкети за събиране на данни и практически експерименти. Проведен е статистически анализ на количествени и качествени данни с цел извличане на значими зависимости и тенденции. Приложена е методология за проектиране на софтуерна система, включваща анализ на изискванията, дизайн, реализация, тестване и внедряване. В процеса на разработка на прототипа са изпълнени тестови сценарии, които оценяват ефективността и приложимостта на разработената система в реална среда, както и е осъществена валидация с експерти от СИ.

1.5. Структура и съдържание на дисертационния труд

Дисертационният труд има следната структура: увод, четири глави, заключение, основни приноси, списък с използвана литература и приложения.

Уводът представя накратко необходимостта от управление на знания в софтуерните компании, мотивацията за избор на тема, дефинира обекта и предмета на изследването, както и основните цели, задачи и приложени методи.

Първа глава представя задълбочен анализ на научната литература и практика, свързани със съвременните подходи и технологии в УЗ и софтуерното инженерство. Разгледани са основните принципи на УЗ и спецификите му в контекста на софтуерните процеси. Дефинирани са ключовите области на знания, които влияят върху ефективността на управлението на софтуерни проекти. В отделна секция са анализирани технологиите, подпомагащи УЗ в софтуерните компании: хранилища за документация, техники за структуриране на информация, колаборативни платформи и системи за обучение. Проведен е сравнителен анализ на различни широкоизползвани системи за управление на документация, като се оценяват техните предимства и недостатъци. Накрая са

очертани новите тенденции в областта като автоматизация, системи за документация и изкуствен интелект, които формират бъдещето на управлението на знания и документи.

Втора глава е посветена на спецификата на УЗ в софтуерните компании, като са систематизирани областите на знания в софтуерното инженерство – знания за продукта и процеса, свързани с контекста и технически и поведенчески знания, свързани с организацията. Проведено е анкетно проучване на потребностите от УЗ в СИ, като е акцентирано върху тенденциите и предизвикателствата, произтичащи от областите на знания.

В третата глава са разгледани възможностите за прилагане на ИИ в управлението на знания и документи в СИ. Представена е концептуална рамка за АСУД, която използва ИИ за подобряване на достъпа, търсенето и управлението на информация в областите на знания. Проведено е проучване за основни функционалности на АСУД, като са анализирани резултатите, които служат като отправна точка за проектиране на системата. Описани са основните ѝ слоеве, управлението на достъпа и интеграцията с ИИ. Следващата част на главата е фокусирана върху разработването на прототип на АСУД, като са дефинирани функционални и нефункционални изисквания, представени са архитектурата и реализацията на системата. Разгледано е и внедряването на прототипа в реална среда.

В четвърта глава е описан процесът на валидация на АСУД, като са изложени методологията, използвана за тестване и оценка на разработения прототип. Разгледани са стъпките по изготвяне на тестови сценарии, които обхващат ключови функционалности на системата, както и подготовката на експеримент за изпълнението на тестовите сценарии. След събирането на количествени данни е представен анализ на резултатите, чрез който е оценена ефективността на разработения прототип.

Заклучението обобщава ползите от предложената концептуална рамка на АСУД, като са изведени постигнатите резултати и са дадени насоки за бъдещо развитие.

2. Проучване на текущото състояние

2.1. Управление на знания и документи

Управлението на знания се разглежда като мултидисциплинарна област, която е тясно свързана с информационните науки (Heisig, 2024). Един от основоположниците на дисциплината УЗ - Карл Уиг (Wiig, K., 2004) счита, че от решаващо значение за жизнеспособността и успеха са активите знания и процесите, свързани със знанията. Знанията трябва да се прилагат, развиват, съхраняват и използват във възможно най-голяма степен от индивидите и организациите, докато процесите, свързани със създаване, кодификация, организиране, преобразуване, трансфер, приложение и защита на знанията, трябва да бъдат внимателно управлявани.

Важен въпрос при УЗ е свързан с намирането на съхранените активи знания. Затова е от съществено значение управлението на съдържанието и използване на метаданни и метазнания, които да подпомагат търсенето. При организирането на знанията се използват таксономии, онтологии, контролирани речници и др. В организационна среда се въвеждат често различни атрибути, например за дейност, област, форма, тип, продукт/услуга, време, местоположение, източник и др.

Ефективната документация е от съществено значение за управлението и споделянето на знания, тъй като осигурява сигурното съхранение на критично важна информация и действа като стратегически инструмент при взимането на решения. Служителите са ценен актив за всяка организация, натрупвайки знания и умения, които трябва да бъдат документирани, за да не се изгубят при тяхното напускане (Navidi et al., 2019). Добре структурираната документация подпомага превръщането на неявните знания в явни, улеснявайки споделянето на опит и добри практики в рамките на организацията. Изследването подчертава предизвикателствата, свързани с документацията, включително липсата на детайли за методологии и грешки, което ограничава възможността за учене от миналия опит.

В този контекст системите за управление на документация (СУД) играят съществена роля в оптимизирането на организационните процеси и насърчаването на сътрудничеството. Според редица изследвания техниките за анализ на настроеността (като класификация на текст и рекурентни невронни мрежи (РНМ)) значително подобряват обработката на текстова информация (Lim et al., 2020). Функционалности като търсене на документи, съобщения и контрол на версиите в СУД се открояват като важни фактори за по-ефективно сътрудничество и подобряване на работните процеси (Regla & Marquez, 2020), което води до увеличено използване на системите (Ayaz and Yanartaş, 2020). Така например, проектирането и внедряването на електронна система за управление на документи (ЕСУД), адаптирана към специфичните нужди на образователни институции, има трансформиращо въздействие върху административните процеси и сътрудничеството (Mahmood, 2017).

2.2. Специфика на УЗ в софтуерното инженерство

Софтуерното инженерство е сравнително нова област от знания, която е претърпяла много промени през последните десетилетия. То възниква преди повече от 50 години като опит да се валидира прилагането на инженерни принципи в софтуерната

разработка, изисквайки специфични умения, които надхвърлят традиционното възприятие за програмирането като процес от писане на алгоритми (Parnas, 2021). Целта е създаването на надеждни софтуерни продукти, подобно на утвърдените инженерни професии като машинното и строителното инженерство.

УЗ в СИ спомага организациите да минимизират рисковете, свързани с изгубени знания, и да се адаптират бързо към промените в технологиите и изискванията на клиентите. В този контекст се въвеждат много инструменти, методи, практики и подходи за създаване на стойност, особено в индустрии, наситени със знания, които предизвикват "експоненциален ръст в публикациите, обхващащи широк спектър от различни и припокриващи се изследователски области" (Arisha et al., 2013). Процесите на програмиране и персонализиране са в основата на тези инструменти, тъй като компаниите трябва да подкрепят генерирането, споделянето и прехвърлянето на знания сред служителите. Това е от решаващо значение, особено в софтуерните процеси, които изискват висока експертиза, тъй като голяма част от знанията, използвани за дадено решение, се приемат като нематериални активи. Стимулирането на тези процеси обаче може да предостави ценна основа за растежа на проекта.

2.3. Области на знания в софтуерните процеси

Терминът област на знанията (knowledge domain) се отнася до конкретна експертиза или специфично поле на знания, което обхваща набор от концепции, правила и информация, свързани с тази област. Областите на знания помагат за организирането и категоризирането на информацията, улеснявайки по-доброто взимане на решения и решаване на проблеми (Cooperman & Karch, 1997).

Областите на знания придобиват различни измерения, когато се разглеждат в контекста на съвременната софтуерна компания. Те се оформят под влиянието, както на вътрешните процеси, свързани с изграждането на организационната култура, така и на външните фактори, включително технологичните иновации и динамичните тенденции в индустриалните практики:

- **Технократски знания** – множество изследвания подчертават необходимостта от по-технократски подход към управлението на знания, например използване на когнитивен анализ, моделиране на събития и клъстериране на данни въз основа на общи характеристики (Osorio et al., 2020).
- **Поведенчески знания** – тази област включва моделиране на екипите с цел идентифициране на специфични поведенчески модели в рамките на проекти. Акцентира се върху човешкия фактор и неговото ефективно въвеждане в софтуерното инженерство чрез образование и мотивация.
- **Продуктови знания** – използват се за трансформиране на софтуерните системи в достъпни, добре описани и модулни услуги (Huang et al., 2008).
- **Процесни знания** – свързани са със знанията за вътрешните процеси и процедури, които софтуерните компании използват за доставяне на своите продукти и услуги.

Всяка една от тези области на знания играе критична роля в ученето и развитието на служителите. Въпреки това, относителната значимост на всяка област може да варира в

зависимост от големината, структурата и опита на компанията. Внедряването на УЗ практики, подпомагани от изкуствен интелект, в компаниите, може значително да оптимизира тези процеси, улеснявайки както обмена на знания, така и тяхното съхранение и прилагане в бъдещи проекти.

2.4. Технологии, подпомагащи УЗ в СИ

Технологиите подпомагат управлението на знания в софтуерното инженерство чрез стимулиране на сътрудничеството, съхранението и анализа на информация. За справяне с предизвикателствата като бързите промени в индустрия, високото текучество на кадри и загубата на знания се използват стратегии като социални мрежи, хранилища за знания, обучения, организационна гъвкавост и работа в екип (Fragoso et al., 2023). Колаборативни инструменти, информационни системи, хранилища на данни и системи за обучения, улесняват споделянето на знания, намаляват риска от загуба на експертиза и насърчават иновациите. Това повишава конкурентоспособността и ефективността на софтуерните компании.

2.4.1. Хранилища на документация

Хранилищата за документация са жизненоважни за достъпността, използваемостта и поддръжката на софтуерни системи, като добрата организация на информацията подобрява ефективността на работните процеси (Alzahrani, 2024). Ефективната документация подобрява комуникацията между екипите, улеснява взимането на информирани решения и подпомага обучението на специалисти. Тя оптимизира ресурсите, подобрява сътрудничеството и насърчава стандартизацията в индустрията.

2.4.2. Техники за документация

Техниките за документация в софтуерните процеси могат да бъдат класифицирани в няколко основни категории (Brody, 2022):

- Статична документация – включва създаването на фиксирани документи, описващи архитектурата, дизайна и функционалността на софтуера.
- Динамична документация – фокусира се върху поведението на системата в реално време.
- Документация, базирана на модели – чрез използване на абстрактни модели се улеснява представянето на сложна информация в разбираем формат.
- Визуална документация – обхваща диаграми, графики и поточни схеми, които допринасят за по-добро разбиране на архитектурата и поведението на системата.
- Колаборативна документация – насърчава съвместното създаване и поддръжка на документацията от различни членове на екипа.

2.4.3. Колаборативни платформи и инструменти

Системите за контрол на версиите (VCS) управляват цифрови ресурси чрез логове на промени, като позволяват връщане към предишни версии, сравнение на редакции и разклоняване на версии за паралелна работа. Те улесняват съвместното редактиране, защитават файловете чрез потребителски права и са фундаментални за проследяване и сливане на данни (Ogayar-Anguita et al., 2023).

DevOps системите от друга страна разширяват принципите на гъвкавата разработка (Agile) чрез непрекъснатата интеграция и доставки, като насърчава автоматизацията на процесите по промяна, конфигурация и пускане на софтуер. Влиянието му върху софтуерното инженерство се изразява в подобро качество, по-бързи итерации и по-добра координация между екипите. Особено важни аспекти са културата, автоматизацията, измерването и споделянето (CAMS), които водят до по-ефективен работен процес и по-надеждни софтуерни решения.

2.4.4. Системи за обучение и обмен на знания

Системите за обучения осигуряват ефективни методи за съхранение, организация и предаване на знания. Тяхната роля е да подобрят уменията и способностите на служителите, съпоставяйки ги със поставените цели в организацията, и да установят култура на продължително учене и споделяне на знания

- Интеграция на електронно обучение (e-learning) с управление на знания – комбинирането на системи за електронно обучение с технологии за управление на знания позволява създаването на гъвкави и ефективни учебни среди.
- Wiki системи за споделяне на знания – използването на Wiki платформи в организационните обучения подпомага създаването и обмена на знания между служителите (Hung & Wang, 2020).

2.4.5. Системи за управление на документация

Системите за управление на документация преминават през няколко поколения на развитие, белязани от въвеждането и усъвършенстването на ключови функционалности.

В периода до 2000 г. (Генерация 1) документацията е предимно статична – състои се основно от печатни ръководства и текстови файлове. Появата на първите дигиталните хранилища стартира дългия процес по дигитализация и индексирание на хартиените документи, като се акцентира върху тяхното форматиране, търсене и архивиране (Kirschen, 1993).

С навлизането на интернет и технологиите от Web 2.0 се появяват интерактивни и базирани на общността платформи за документация (Генерация 2 - до 2010 г.). Целта е да се осигури по-добро сътрудничество между различните екипи посредством споделен достъп, управление на версиите и нотификации.

Следващият период до 2000 г. (Генерация 3) бележи интеграцията на платформи за сътрудничество в реално време като Slack и Discord, които позволяват на разработчиците да обсъждат и документират процеса на работа по-динамично (Raglianti, 2023). Това дава началото на нови подходи в УЗ, които имат за цел да повишат качеството и използваемостта на наличната документация (Ding et al., 2014).

След 2020 г. (Генерация 4) СУД стават все по-разнообразни, интегрирайки различни хранилища, интерактивни уроци и платформи за незабавни съобщения. ИИ и машинното обучение започват да се използват за автоматично генериране и актуализиране на документация, което повишава нейната точност и съответствие със софтуерните промени (Moghaddam et al., 2022).

2.5. Сравнителен анализ на системите за управление на документация

В таблица 2.1 са сравнени някои от популярните технологии според изброените фактори.

Таблица 2.1. Сравнителен анализ на системи за управление на документи

| Фактор | Microsoft SharePoint | Confluence (Atlassian) | GitHub Wiki | DocuSaurus |
|--------------------|--|---|--|---|
| Тип на хранилището | Корпоративно, централизирано | Колаборативно, облачно | Децентрализирано, Git-базирано | Генерирано статично съдържание |
| Основни функции | Управление на документи, съвместна работа, търсене, контрол на версиите | Сътрудничество, документен мениджмънт, интеграция с Atlassian екосистемата | Поддържана в GitHub документация, Markdown поддръжка | Статична документация, Markdown, React-базирано съдържание |
| Сигурност | Високо ниво на сигурност, ролеви модели за достъп, криптиране | Гранулиран контрол на достъпа, но не толкова детайлен, колкото при SharePoint | Зависи от настройките на GitHub, ограничен контрол върху достъпа | Зависи от хостинг платформата, без вграден контрол на достъпа |
| Интеграции | Microsoft 365, Teams, OneDrive, Power Automate | Jira, Trello, Bitbucket, Slack | GitHub, CI/CD инструменти | GitHub, Netlify, Vercel, CI/CD процеси |
| Предназначение | Големи компании, нуждаещи се от сигурно управление на документи и сътрудничество | Технологични екипи, търсещи гъвкава и добре интегрирана документация | Проекти с отворен код и GitHub-базирани работни процеси | Техническа документация, особено за API-та и библиотеки |

2.6. Нови тенденции в управлението на знания и документи

С развитието на технологиите за ИИ се предоставят нови възможности за трансформиране на процесите, свързани със споделянето на знания и управлението на документация (Jarrahi et al., 2023). Благодарение на напредъка в обработката на естествен език, ИИ може да анализира и обработва огромни масиви от неструктурирани данни – документи, имейли, чатове, презентации и други текстови източници – като извлича най-съществената информация и автоматично я категоризира.

Създаването на разбираема документация чрез ИИ гарантира съответствие с регулаторните стандарти (Königstorfer & Thalmann, 2022). Различни изследователи

акцентират върху съществената роля на документацията, която осигурява човешки контрол върху системите с ИИ и подпомага оценката на въздействието на човешките решения върху бизнес процесите.

Внедряването на решения с ИИ води до по-ефективно управление на знания, като не само автоматизира рутинните процеси, но и намалява риска от загуба на ключова информация. В дългосрочен план тези технологии ще играят централна роля в трансформирането на работния процес, като осигуряват по-добра свързаност, достъпност и адаптивност на знанията в една организация.

2.7. Изводи

Проучването на съществуващите подходи за УЗ и документи в контекста на СИ разкрива редица предизвикателства и области, в които е необходимо да се направят изследвания. Анализът на методите за управление на знания и документи показва, че съществуват различни технологични решения, чрез които се подпомага ефективната интеграция на областите на знания в СИ. Хранилищата за документация, техниките за документирание и колаборативните инструменти показват напредъка в сферата на автоматизацията, но въпреки предоставените възможности за структуриране и обмен на информация, съпоставянето на различните инструменти подчертава, че всеки от тях има своите ограничения по отношение на мащабируемостта, интеграцията с външни системи и степента на автоматизация при извличане и кодифициране на знания.

Наред с това, идентифицираните нови тенденции в областта на управлението на знания и документи подчертават нарастващата роля на изкуствения интелект, машинното обучение и автоматизираните анализи като средства за подобряване на достъпността, структурираността и ефективността на информационните системи. Въпреки това, интеграцията на тези технологии с традиционните методи за управление на знания остава недостатъчно проучена и изисква допълнителни изследвания, насочени към разработване на интелигентни системи за управление на знания в сложни софтуерни екосистеми.

Тези изводи очертават необходимостта от по-задълбочено разглеждане на спецификата на управлението на знания в софтуерното инженерство, което ще бъде предмет на следващия етап от изследването. Фокусът е поставен върху разрастването на областите от знания в софтуерните процеси, както и върху начините за тяхното ефективно структуриране, извличане и използване в практическа среда.

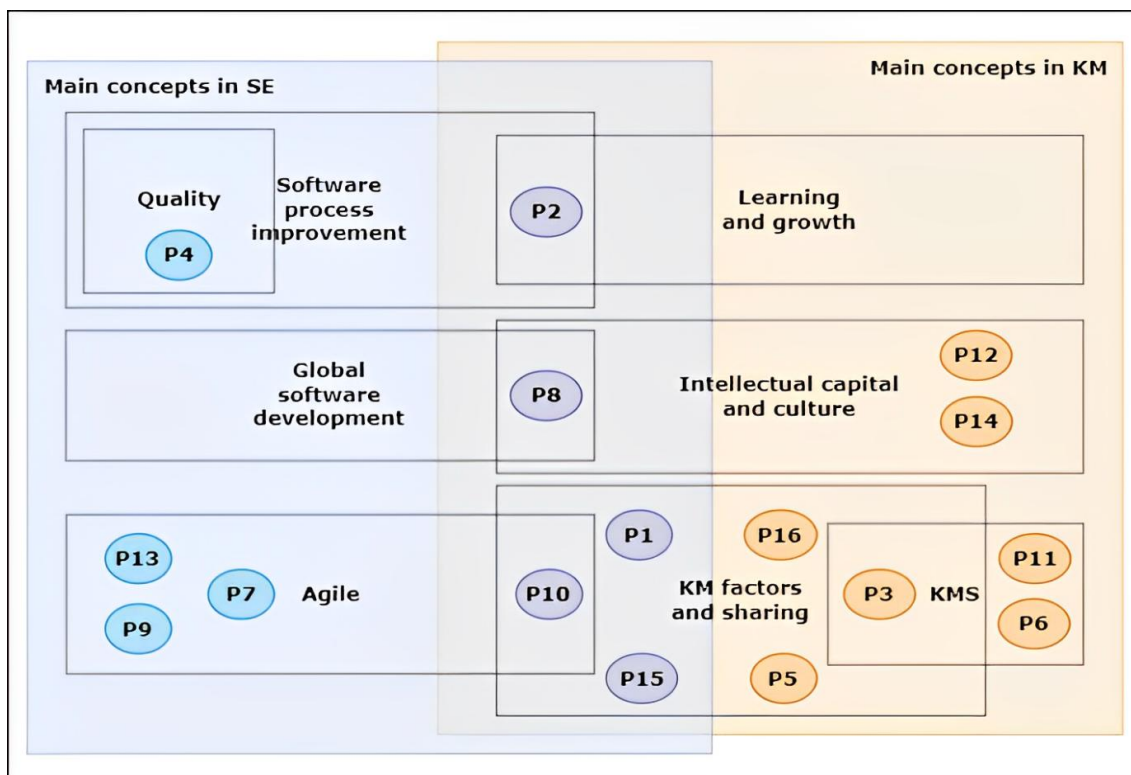
3. Проучване на управлението на знания в софтуерните компании

3.1. Систематизиране на областите на знания в софтуерното инженерство

За да се дефинират областите на знанията в СИ, е създадена опростена концептуална визуална схема, която включва 16 систематични прегледа (P1-P16), представени с ключовите им думи и основните им теми и реферирани в по-нататъшните съждения посредством зададените им идентификатори. На база ключовите думи и изводите от реферираните проучвания, както и чрез сравняване на различни аспекти е направен анализ на връзките между областите, обединени в две основни концептуални тематични групи – за софтуерно инженерство и за управление на знания. Тези групи оформят теоретичната основа на проучването, като в тях се идентифицират съпоставими теми, които осигуряват детайлно разбиране на водещите процеси, методологии и фактори, свързани с развитието на софтуерната индустрия и ефективното управление на знания.

На фигура 3.1. е представено разпределението на различните източници според концепциите в УЗ и СИ:

- Концепции за СИ:
 - Управление на качеството – P4;
 - Гъвкави методологии – P7, P9, P13;
- Концепции за УЗ:
 - Интелектуален капитал и култура – P12 и P14;
 - Фактори при УЗ – P3, P5, P6, P11, P16;
- Общи концепции на УЗ и СИ – P1, P2, P8, P10, P15.



Фигура 3.1. Визуална схема на основни концепции в СИ и УЗ идентифицирани в данните на изследването

3.2. Знания за контекста в Софтуерното инженерство

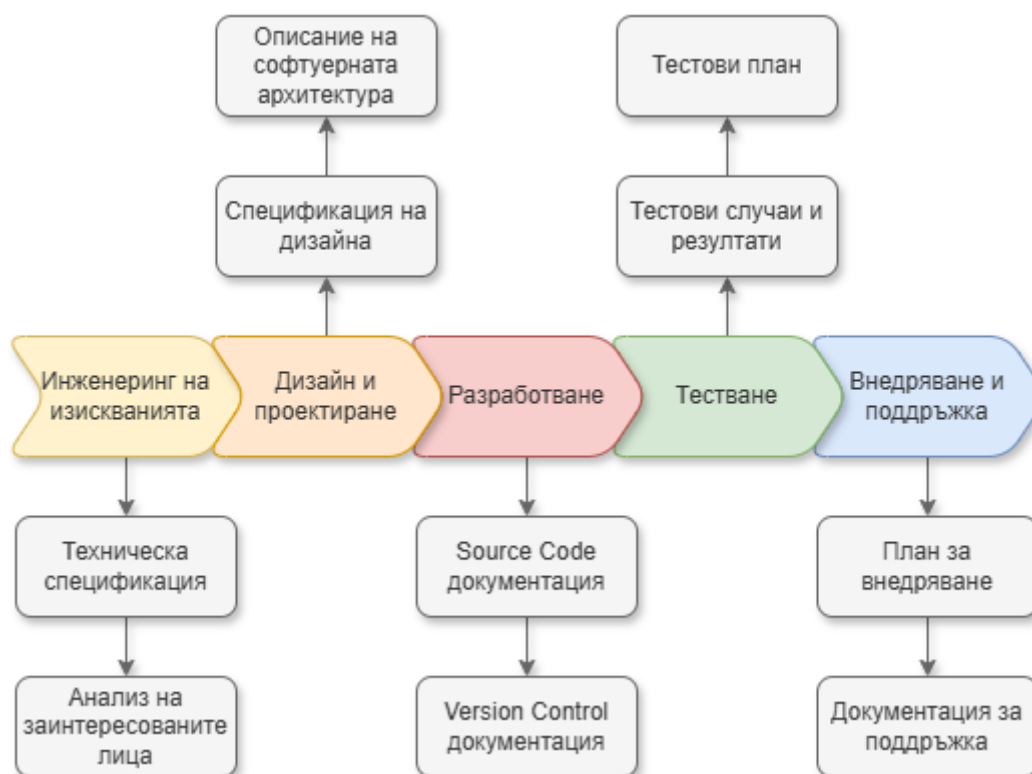
В софтуерното инженерство контекстът представлява съвкупността от фактори, които влияят върху проектирането, развитието и работата на една софтуерна система. Разбирането на този контекст е от съществено значение за създаването на ефективни, надеждни и лесни за използване решения (Bedjeti et al., 2017).

3.2.1. Знания за продукта

3.2.1.1. Документация за продукта

В хода на жизнения цикъл на софтуерната разработка тези знания се създават, изменят и натрупват, добавяйки изисквания както към същността на разработвания продукт, така и към технологиите и формата на данните.

Жизненият цикъл на софтуерната разработка (SDLC) преминава през няколко фази, като всяка от тях изисква специфична документация, осигуряваща успешното изпълнение на проекта. Основните документи в SDLC варират според фазата (Фигура 3.2.), като предоставят критична информация за всички заинтересовани страни и разработчици (Shetty et al., 2023) (Aversano et al., 2017):



Фигура 3.2. Документи в жизнения цикъл на софтуерната разработка

3.2.1.2. Подходи при описанието на продукта

Съществуват различни подходи за описване на продукта чрез машинно четими спецификации, обогатени с метаданни и механизми за откриване на конфликти. Те подпомагат трансформацията на остарели системи в достъпни, добре описани и съвместими модулни услуги, ускорявайки промените при гъвкава софтуерна доставка.

Описателни езици като UML, VEL и WADL играят роля в дизайнерски подходи като QDAD, Q-Rapids и Transformation models. Мета-базираните модели анализират софтуерния дизайн и бизнес процесите, категоризират данни според типа, използваемостта и консистентността, подпомагайки оценката на усилията и прогнозите.

Моделирането на съответствие идентифицира предизвикателства като липсваща връзка с бизнес случаите и създава нови теории за формализация на процесите и ангажирането на заинтересованите страни.

Класификационни методи като геймификация, прототипиране и CASE подпомагат процесите чрез подобряване на екипната ефективност и управление на техническия дълг.

Продуктово-ориентираните компании използват мрежи от знания и обратна връзка за подобрене на версиите, докато User-Centred Designs адаптират софтуерните продукти към различни потребители. Подходи като Product Road Mapping и Visual Milestone Planning подпомагат приоритизацията и откритието на продукти в индустрии като видео игрите.

3.2.2. Знания за процеса

Събирането на знанията, които даден софтуерен процес генерира, е предизвикателство за всеки ръководител на екип. Идентифицирани са различни методи, които подкрепят тази дейност по събиране на знания, като се разделя жизненият цикъл на софтуерната разработка (SDLC).

По време на фазата на изпълнение, фокусът е върху предоставената стойност и начина, по който се изпълнява процесът на разработка – водопаден (Waterfall), итеративен (Iterative), V-образен (V-Shaped) или спираловиден (Spiral).

По-късно, по време на фазата на оценка на качеството, се вземат предвид нефункционални изисквания като сигурност, поддръжка, прозрачност и издръжливост.

3.3. Организационни знания в софтуерното инженерство

На първо място, трябва да се направи важно разграничение между концепциите за УЗ и СИ. Когато се разглеждат организационни теми, трябва да се вземат предвид организационната култура и структура, както и мисията, визията, ученето и растежът. Организационната структура подпомага трансфера на неявни знания през различните управленски нива (Naq & Anwar, 2016). От друга страна, подкрепяната от компанията култура е от решаващо значение за споделянето на знания между екипи и отдели, както и за мотивацията на служителите. Мисията и визията определят общата насока на развитие на организацията, а ученето и растежът могат да допринесат за въвеждането на нови възможности и предизвикателства.

3.3.1. Технократски знания

Основните изследвания посочват нуждата от по-технократски подход към управлението на знания. Това включва когнитивен анализ, по-задълбочен поглед върху комуникацията във виртуална среда, моделиране на събития и клъстеризиране на данни, базирани на общи характеристики (Osorio Angel et al., 2019). Това е съществено

за управлението на проекти, тъй като кодификацията и документирането са жизненоважни дейности при мониторинга (Ayarza & Bayona-Oré, 2020).

Внедряването на информационни системи създава една динамична среда, която има нужда от непрекъснат контрол. За тази цел технологии и методи за одит на знания помагат на организацията за картографиране на активи знания, разработване на карта на средата на знания, моделиране на потоците знания, анализ на социалната мрежа, конкурентен анализ, анализ на критични функции и други (Gourova & Dragomirova, 2015).

Дизайнът на СУЗ в този контекст трябва да се обмисли в контекста на обхвата, методологията на разработка, човешките ресурси, допълнителните активи, практиките за управление на знания и елементите на инфраструктурата (Gourova & Toteva, 2014). Друг важен въпрос, който трябва да се обсъди, е процесът на мониторинг, който трябва да осигури абстрактни, мета-ориентирани изгледи, отразяващи организационната структура (Torres et al., 2019).

3.3.2. Поведенчески знания

По отношение на поведенческия подход, трябва да се анализират две основни направления: едното е свързано с моделирането на екипи и откриването на шаблони (patterns) в по-големи проекти, а другото се базира на Agile рамката и как тя може да трансформира по-големи компании в силно свързани общности на знанията.

Изследванията показват, че човешките фактори са обект на анализ и според литературни прегледи основните клъстери описват как те се въвеждат ефективно в процеса на софтуерното инженерство и как го подобряват чрез образование и мотивация (Machuca-Villegas et al., 2020). Подобен анализ, приложен в голям мащаб, идентифицира нуждата от одит на съществуващите методи и практики, така че екипите да бъдат по-добре подготвени и адаптивни към всякакъв контекст. Например, глобалната софтуерна разработка се фокусира върху по-абстрактни принципи, дефиниращи софтуерната разработка като „човеко-центрична и социо-техническа активност“ и вземаща предвид културния контекст при управлението на по-големи проекти (Marinho et al., 2018).

3.4. Проучване на потребностите от управление на знания в софтуерните процеси

Текущото проучване се прави с цел да се изследват областите на знания в СИ, като се поставят въпроси за наличността и нивата на достъп до знанията в различните софтуерни компании, използваните системи за осъществяване на достъп, отношението на служителите към гъвкавите методологии и организационната култура, както и трудностите, които служителите изпитват в хода на работата си.

3.4.1. Методология на проучването

Социологическите проучвания са мощен инструмент за събиране на данни, които да послужат за формирането на изводи за нагласите и тенденциите в дадена индустрия. В текущото изследване целта е да се проучат потребностите, за което подходяща методология би било проучване от описателен характер (Babbie, 2020).

Моделът за описателния тип проучвания се базира на дефинирането на въпроси, произтичащи от дадена хипотеза или теория (Creswell, 2014). Въпросите имат за цел да измерят отношението на респондентите по дадени теми, което предразполага оформянето им като анкети, както със затворени отговори (Ликертова скала, множествен избор), така и с отворени такива, където респондентите могат да дадат допълнителна информация.

За инструмент за планираното проучване е избран въпросникът. Взето е предвид, че могат да се използват отворени или затворени въпроси, при което първите служат за получаване на допълнителна информация от респондентите, както и подпомагат евентуални следващи дълбочинни проучвания, но по принцип по-трудно се обработват. Затворените въпроси съдействат за определянето на мнението и отношението на респондентите към различни проблеми, за които те дават оценка по определена скала (от 1 до 5).

За да се анализират резултатите от анкетата, е приложен едновибрационен t-тест на Стюдънт. Целта на анализа е да се провери дали средните стойности на отговорите по определени твърдения статистически се различават от неутралната стойност на скалата, обикновено фиксирана като 3 – нулева хипотеза $H_0: \mu=3$. Това позволява оценка за това дали респондентите изразяват склонност към съгласие (хипотеза $H_1: \mu>3$) или несъгласие (хипотеза $H_1: \mu<3$) по ключови теми, свързани с различните области на знания в СИ.

3.4.2. Анализ на резултатите

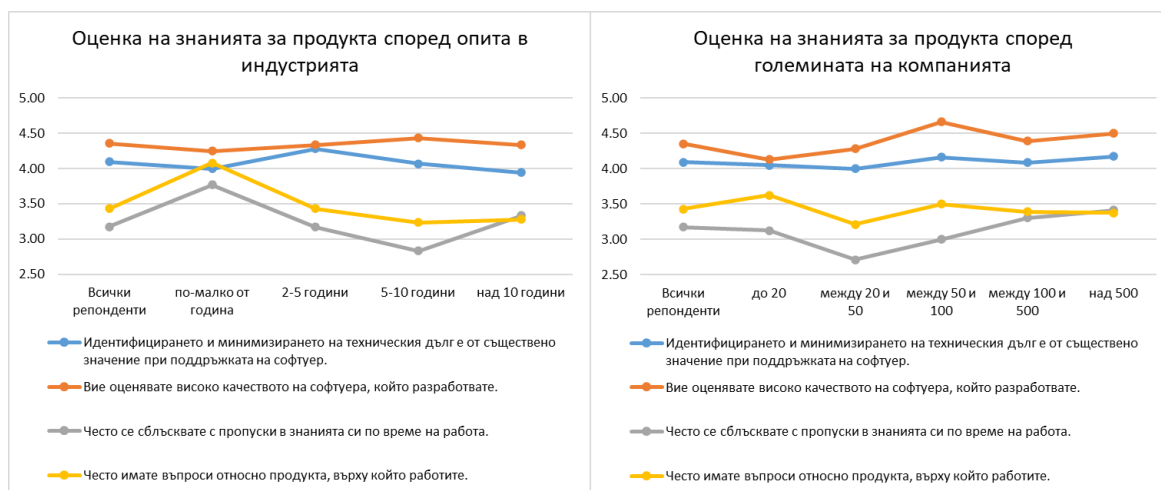
Въпросникът е разпространен чрез Google Forms, включва 48 въпроса и е изпратен до компании, експерти и студенти. Анкетата е проведена януари – февруари 2023 г., събрани са 91 отговора от професионалисти в софтуерното инженерство. Анализът отчита фактори като опит, фирмен размер и структура, идентифицирайки основни тенденции и предизвикателства.

90% от анкетираните имат постоянни позиции в софтуерната индустрия. 73% са разработчици, 25% заемат управленски роли, а 42% изпълняват повече от една длъжност. 53% имат над 5 години опит, като 20% работят в сектора над 10 години, а само 14% са с под 1 година опит.

51% са в компании с над 100 служители, 26% в малки фирми с под 20 души.

Основната част от проучването изследва четирите основни области на знания, като се фокусира върху продуктите и процесите в СИ, както и върху организационните аспекти на СИ от технократски и поведенчески стилове. За анализ се използва скала на Ликърт (1 – категорично не, 5 – категорично да, 3 – неутрално).

Графиките на фигура 3.3 представят сравнение на въпросите свързани с продукта. Включени са t-тестове, оценяващи твърденията за знанията за продукта според отделните групи (Таблица 3.1) – t-тестове са приложени и за другите области на знания, като изводите са приложени в текущия реферат.



Фигура 3.3. Оценка на знанията за продукта според опита и големината на компанията

Чрез сравнителен анализ могат да се направят следните изводи:

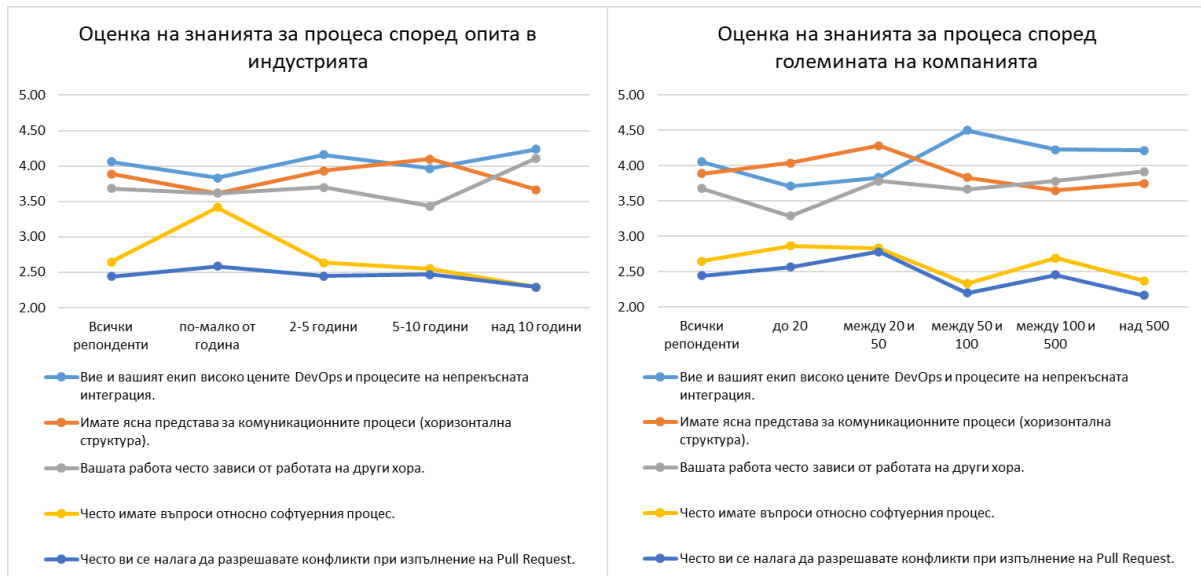
- Качество на софтуера – Участниците оценяват високо софтуерния продукт, независимо от опит или размер на компанията, което показва зрялост в процесите на валидация и верификация.

Таблица 3.1. Оценка на знанията за продукта според опита и големината на компанията (t-тестове)

| Въпрос | Фактор | Група | Средна стойност | t-стойност | p-стойност | Резултат |
|--|------------------------|--------------------|-----------------|-------------|-------------|-----------|
| Идентифицирането и минимизирането на техническия дълг е от съществено значение при поддръжката на софтуер. | общо | | 4.10 | 11.69 | 0 | $\mu > 3$ |
| | години опит | по-малко от година | 4.00 | 3.32 | 0.003436152 | $\mu > 3$ |
| | | 2-5 години | 4.28 | 8.68 | 3.59525E-09 | $\mu > 3$ |
| | | 5-10 години | 4.07 | 7.06 | 4.57987E-08 | $\mu > 3$ |
| | | над 10 години | 3.94 | 0.00 | 0.000499243 | $\mu > 3$ |
| | големина на компанията | до 20 | 4.05 | 6.19 | 3.03496E-06 | $\mu > 3$ |
| | | между 20 и 50 | 4.00 | 3.61 | 0.001804479 | $\mu > 3$ |
| | | между 50 и 100 | 4.17 | 3.80 | 0.006338301 | $\mu > 3$ |
| | | между 100 и 500 | 4.09 | 5.55 | 8.18633E-06 | $\mu > 3$ |
| над 500 | | 4.17 | 6.35 | 1.09019E-06 | $\mu > 3$ | |
| Вие оценявате високо качеството на софтуера, който разработвате. | общо | | 4.36 | 16.42 | 0 | $\mu > 3$ |
| | години опит | по-малко от година | 4.25 | 5.74 | 6.47008E-05 | $\mu > 3$ |
| | | 2-5 години | 4.33 | 7.62 | 1.06904E-08 | $\mu > 3$ |
| | | 5-10 години | 4.43 | 11.56 | 1.108E-12 | $\mu > 3$ |
| | | над 10 години | 4.33 | 8.25 | 1.20492E-07 | $\mu > 3$ |
| | големина на компанията | до 20 | 4.13 | 6.24 | 1.3985E-06 | $\mu > 3$ |
| | | между 20 и 50 | 4.29 | 4.50 | 0.000298645 | $\mu > 3$ |
| | | между 50 и 100 | 4.67 | 5.00 | 0.002052358 | $\mu > 3$ |
| | | между 100 и 500 | 4.39 | 11.44 | 4.91199E-11 | $\mu > 3$ |
| над 500 | | 4.50 | 11.14 | 4.76416E-11 | $\mu > 3$ | |

- Неясноти относно продукта – Честите въпроси сигнализират за липса на документация или динамични изисквания. По-опитните специалисти по-бързо се адаптират и задават по-малко въпроси.
- Технически дълг – Екипите осъзнават рисковете от неефективен код и архитектурни компромиси, подчертавайки важността на минимизирането му.
- Пропуски в знанията – Те се дължат на недостатъчно ефективно споделяне на информация или липса на интегрирани системи за управление на знанията.

Графиките на фигура 3.4 представят анализ на въпросите свързани с процеса, като се наблюдават следните изводи:



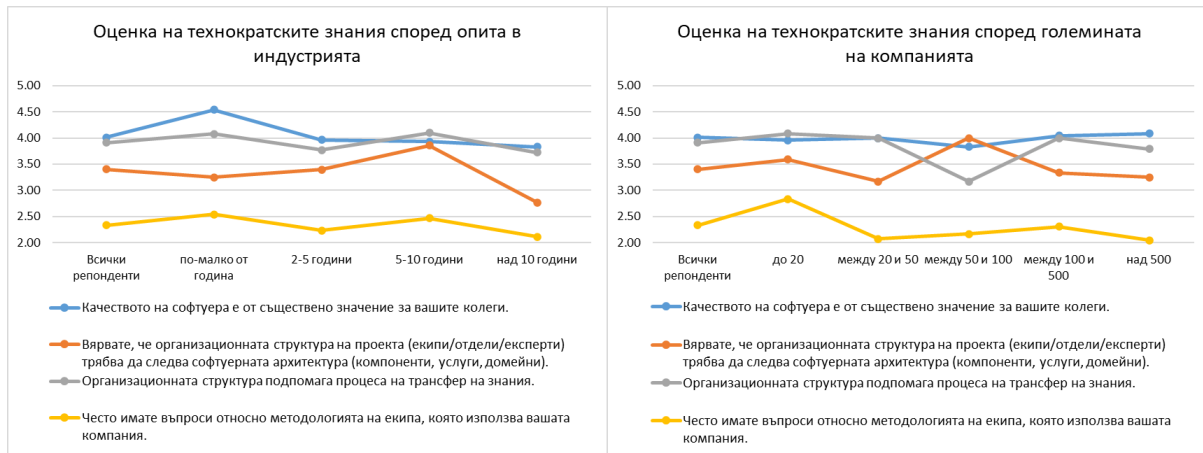
Фигура 3.4. Оценка на знанията за процеса според опита и големината на компанията

Чрез сравнителен анализ могат да се направят следните изводи:

- DevOps и CI/CD – Компаниите разчитат на автоматизация и бърз цикъл на разработка. По-големите фирми (50+ души) оценяват тези процеси по-високо (4.5) заради сложността на проектите и нуждата от множество тестови среди.
- Комуникация в екипа – Добрата координация е ключова, с най-висока оценка (4.29) при фирми от 20–50 души. В по-големите компании комуникацията се усложнява.
- Зависимост от други хора – Нараства с размера на компанията – от 3.29 при малките фирми (до 20 души) до 3.92 при големите (500+), поради специализацията на ролите.
- Въпроси за процесите – По-малките организации разчитат на неясни знания, докато големите имат по-добра документация. С опита неяснотите намаляват.
- Конфликти при сливане на код (Pull request) – Срещат се рядко, което показва добро управление на версиите и синхронизация между екипите.

Графиките на фигура 3.5 представят анализ на въпросите свързани с технократските знания, като се наблюдават следните изводи:

- Качеството на софтуерните продукти – служителите ценят надеждността и добрите практики в разработката.
- Организационна структура – важна за екипните роли и начина по който влияе върху комуникационния процес.
- Трансфер на знания – улеснен от организационната структура.
- Отношение към методологиите – има възприети добри практики и известна яснота.



Фигура 3.5. Оценка на технократските знания според опита и големината на компанията

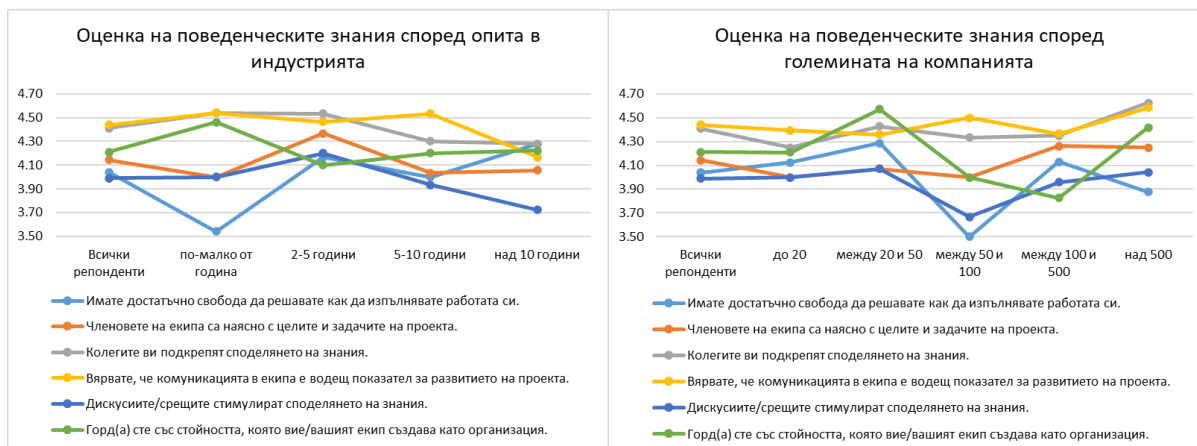
Основни предизвикателства са липсата на менторство (70%), недостатъчните познания за нови технологии (42%) и липсата на документация (29%). Компаниите насърчават обмена на знания хоризонтално (70%) и вертикално (58%) чрез обучения и менторски програми.

Анализът на поведенческите знания показва (фигура 3.6):

- Свобода на действие – служителите имат автономност, но средните по размер екипи (50-100 души) се чувстват най-ограничени.
- Осъзнаване на целите – големите екипи и служителите с 2-5 години опит разбират най-добре задачите.
- Споделяне на знания – активно практикувано, ключово за организационното развитие.
- Комуникация – възприемана като водещ фактор в екипната работа.
- Дискусии – стимулират обмена на знания и подобряват координацията.
- Екипна стойност – най-високо оценена от компании с 20-50 души (4.6), най-ниско от 100-500.

Неявните знания играят ключова роля в адаптацията на новите служители, като основни източници са колегите (75%), лидерите (48%) и мениджърите (21%). Липсата на ефективна комуникация между управленските нива (60%) затруднява този процес. Компаниите отговарят с инициативи за сплотеност и култура на споделяне.

Основен проблем е трудността при споделяне на знания за сложни задачи, което води до забавяния, грешки и загуба на знания, особено при ротация на служители. Това се наблюдава както в малки, така и в големи компании. Подобни затруднения ограничават иновациите и ефективността.



Фигура 3.6. Оценка на поведенчески знания според опита и големината на компанията

Въпреки наличието на съвременни инструменти за управление на знания, ефективността им не винаги е висока. Опитните служители се адаптират и кодифицират знания, докато по-неопитните търсят помощ от документация и ментори. В международни екипи културните различия допълнително усложняват обмена.

Организационната култура е решаваща за ефективността и споделянето на знания. Подходите към обучение и комуникация влияят върху изграждането на сплотени и продуктивни екипи. УЗ трябва да подпомага адаптацията и обмена на практически знания.

ИИ и автоматизацията предлагат решения за съществуващите предизвикателства, подобрявайки продуктивността чрез по-бързо намиране и споделяне на информация. Това показва, че иновациите могат успешно да се интегрират в управлението на знания.

3.5. Изводи

Успешната интеграция на УЗ е задължителен елемент от ефективното изпълнение на процесите в СИ, стъпвайки както върху прилагането на гъвкави методологии, така и върху автоматизацията на тестването и внедряването на софтуерните продукти.

Един от ключовите проблеми е отсъствието на структурирани менторски програми. Проучванията показват, че 75% от новите служители разчитат основно на колегите си за усвояване на знания.

Достъпът до актуална и добре структурирана документация е от съществено значение за продуктивността на екипите. Въпреки това, 29% от анкетираните отбелязват, че документацията не е достатъчно актуална или лесно достъпна. Допълнително, 60% от служителите подчертават, че комуникацията между различните управленски нива е затруднена, което води до забавяне на процесите и неефективност в координацията.

За да се преодолеят тези предизвикателства, е необходимо въвеждането на стратегически подходи и технологични решения, които да подобрят управлението на знания и оптимизират работните процеси.

Създаването на официални менторски програми и вътрешни платформи за споделяне на опит ще подпомогне адаптацията на новите служители. Чрез тези инициативи

новопостъпилите ще имат достъп до целенасочени ресурси и насоки, което ще ускори интеграцията им в екипа.

Интегрирането на системи като СУД ще позволи по-ефективно управление на техническата документация, като осигури централизирано хранилище за актуална информация. Това ще намали натоварването върху разработчиците и ще улесни достъпа до необходимите данни.

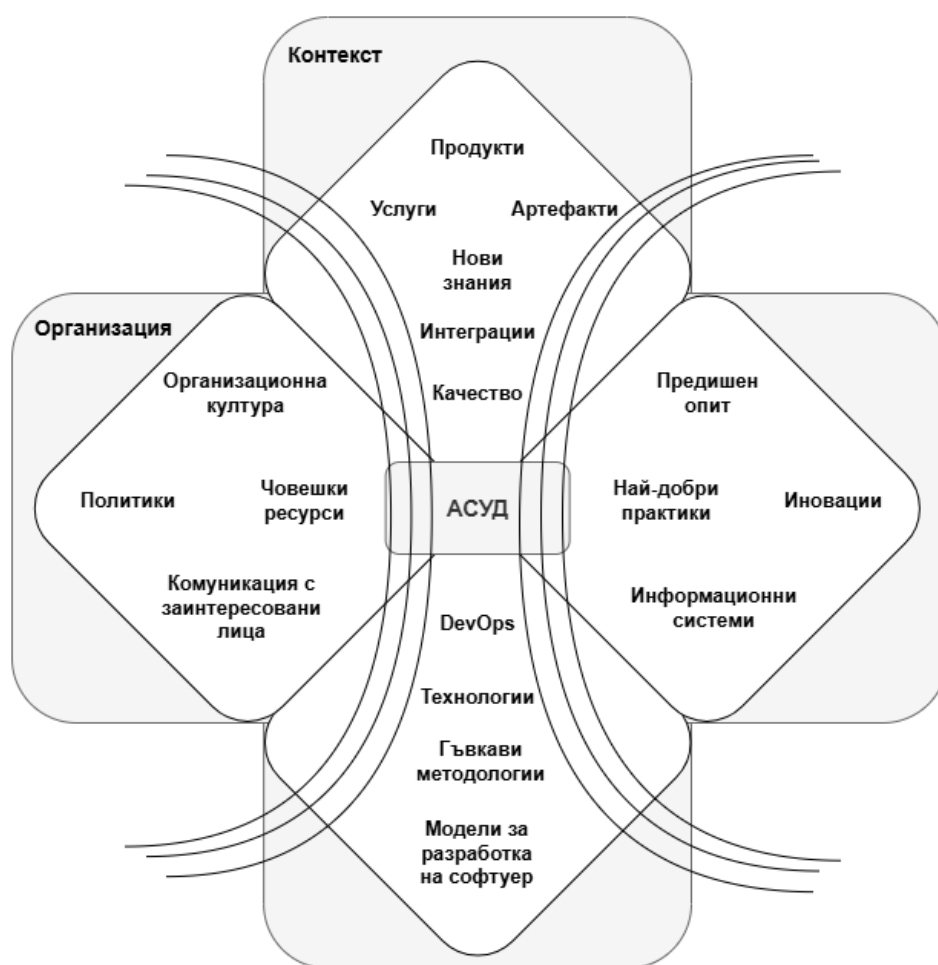
Прилагането на тези мерки не само ще подпомогне текущите работни процеси, но и ще допринесе за дългосрочното развитие на компаниите. Чрез ефективно управление на знанията и оптимизиране на технологичните ресурси организациите ще могат да подобрят продуктивността си, да намалят времето за внедряване на иновации и да осигурят по-добра среда за своите служители.

4. Приложение на изкуствен интелект при управление на знания в софтуерните процеси

4.1. Концептуална рамка за Автоматизирана система за управление на документи

Управлението на документация е предизвикателство поради труднодостъпната, неструктурирана и усложнена информация. Това води до затруднена интеграция на нови служители, по-ниска ефективност на работните процеси и намалена скорост на внедряване на иновации.

Целта на АСУД в този контекст е да осигури централизиран и интелигентен подход към управлението на техническата документация чрез използване на ИИ и систематичен модел на УЗ в СИ. Системата трябва да адресира четирите основни области на знания в СИ (Фигура 4.1):

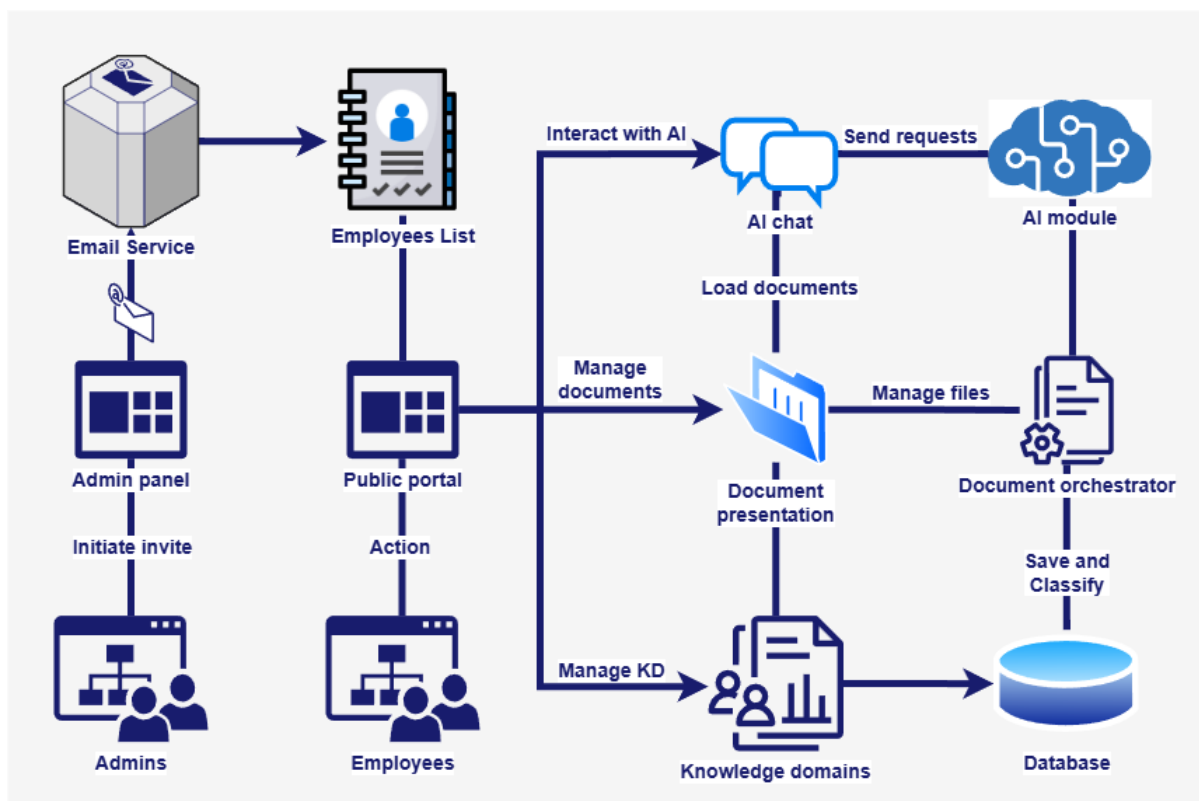


Фигура 4.1. Мястото на АСУД спрямо различните области на знания

На Фигура 4.2 е илюстрирана концептуалната рамка за АСУД, като се акцентира върху два основни компонента:

- Административен панел, който контролира достъпа и правата на редовните потребители;
- Портал, който според делегираните права на потребителя да го пренасочва към различните области на знания, където да се осигури достъп до документацията,

като се предоставят както инструменти за анализ на знания с ИИ, така и табла с обобщена информация и препоръки за ефективно използване.



Фигура 4.2. Концептуална рамка за Автоматизирана система за управление на документи

4.2. Проучване за основните функционалности в АСУД

4.2.1. Методология на проучването

Целта на текущото изследване е да се проучат нагласите на експерти в областта по отношение на функционални и нефункционални изисквания, засегнати в концептуалната рамка на АСУД. Поради тази причина е избрана методология на проучване от аналитичен характер (Babbie, 2020). Такова проучване подпомага първоначалната концептуална рамка, като приоритизира изначално дефинираните в модела функции.

В продължение, за да се придобие представа за мащабите и контекста, са взети предвид следните характеристики:

- Опит – брой години и направление на респондента;
- Големина на компанията, в която работи респондентът.

4.2.2. Анализ на резултатите

През март-април 2024 г. е проведено проучване сред ИТ специалисти в България с цел оценка на функционалните и нефункционалните изисквания към системите за управление на документи. Анкетата, разпространена сред служители на софтуерни компании, използва скала от 1 до 5, като респондентите имат възможност да отбележат изискванията като неприложими.

Изследването включва мнения от разнообразни професионалисти, като 55% от респондентите са софтуерни разработчици, следвани от 20% специалисти по осигуряване на качеството, 17% ръководители на екипи, 11% проектни мениджъри, 11% специалисти за човешките ресурси (ЧР) и 8% софтуерни архитекти. Всеки участник има минимум две години опит в своята сфера. Почти 45% от респондентите работят в компании с между 100 и 500 служители, докато 30% са заети в организации с над 500 души персонал.

Представена е общата оценка на различни функционалности на АСУД:

- Функции като търсене на документи и история на документите получават високи средни стойности от 4.2 (от 5), което предполага, че потребителите ценят стабилните възможности за търсене и инструментите, които подпомагат създаването и редактирането на съдържание.
- Съхранението на документи, импортирането/експортирането и споделяният достъп също получават положителна обратна връзка с оценки близки до 4, което подчертава значението на събирането и прехвърлянето на знания, както и осигуряването на достъпност.
- Чат инструменти и интеграция на външни плъгини с ИИ, получават сравнително по-ниски оценки (3.2).

Представени са оценките за различни нефункционални изисквания към системите за управление на документи, като всяко изискване отразява ключови аспекти, които влияят върху производителността на системата и удовлетвореността на потребителите.

- Ефективност – оценена с 3.8, което показва, че потребителите възприемат предложените функционалности като ефективни при изпълнение на задачите, свързани с управлението на документи.
- Сигурност – с оценка 3.2, което подсказва необходимост от подобрене в защитата на данните и осигуряването на конфиденциалност.
- Адаптивност – с рейтинг 3.6, което показва умерена способност на предложения прототип да се адаптира към промени и специфични предпочитания на потребителите.
- Интегритет – оценен с 3.0, което предполага, че потребителите възприемат прототипа като относително цялостен, но с възможности за подобрене в осигуряването на по-стабилна и интегрирана функционалност.

Тези резултати предоставят ценна информация за организациите, като посочват ключови области за подобрене на системите за управление на документи.

4.3. Проектиране на АСУД

АСУД представлява решение за цялостно управление на разнообразна документация, обхващаща човешки ресурси, организационни процедури, портфейли и продукти в рамките на една организация (Georgiev, 2023).

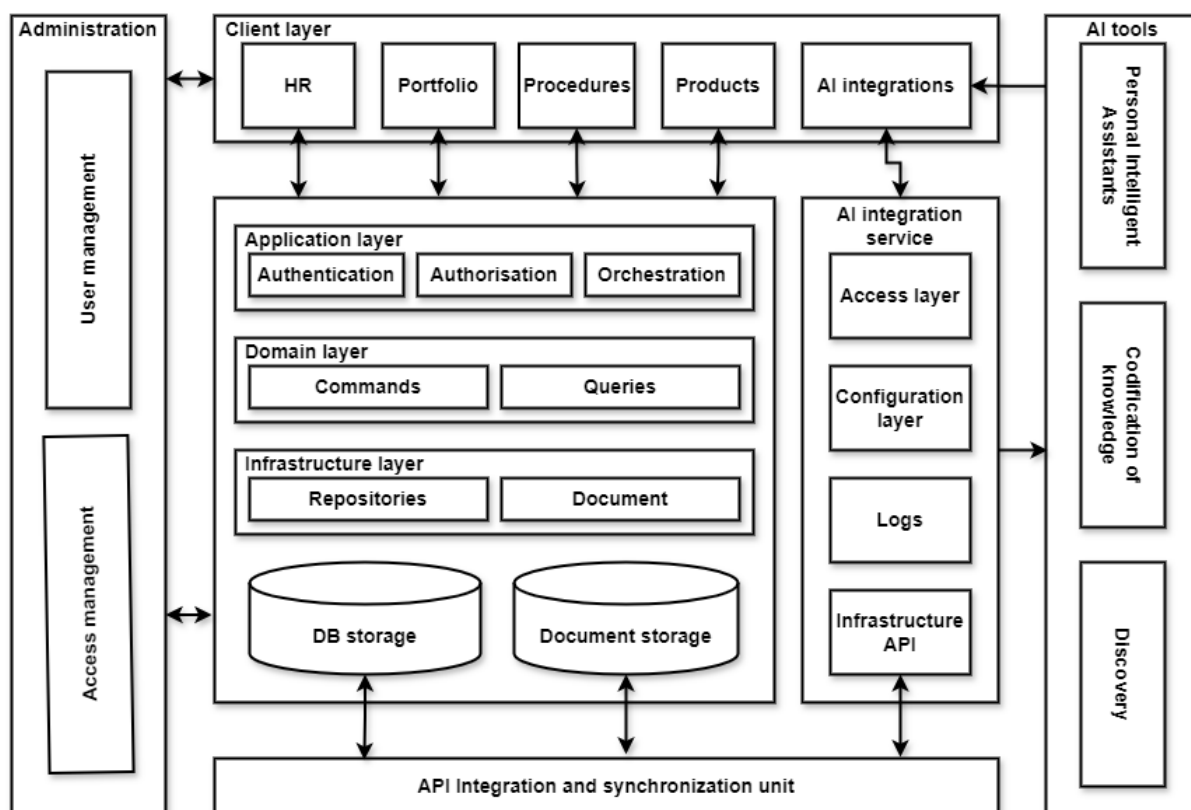
Архитектурният модел АСУД е многослойна структура, интегрираща различни компоненти за ефективно управление на знания, достъп до документи и интеграция с изкуствен интелект (Фигура 4.3).

4.3.1. Основни слоеве и компоненти на системата

4.3.1.1. Клиентски слой (Client layer)

Клиентският слой предоставя Web базирана апликация, чрез която потребителите взаимодействат със системата. Тя включва няколко модула:

- Човешки ресурси;
- Портфолио;
- Процедури;
- Продукти;
- ИИ интеграции;



Фигура 4.3. Архитектурен модел на АСУД

4.3.1.2. Приложно ниво (Application layer)

- Автентикация – удостоверяване на потребителите при достъп;
- Упълномощаване – контрол на правата за достъп;
- Оркестрация – управление на бизнес логиката и комуникацията между компонентите.

4.3.1.3. Доменен слой (Domain layer)

- Commands (Команди) – изпълнение на заявки за промяна в системата.
- Queries (Запитвания) – извличане на информация от базата данни и документалните хранилища.

4.3.1.4. Инфраструктурен слой (Infrastructure layer)

- Модул за осигуряване на достъп до хранилища (Repository) – осигурява връзката с базата данни, изпълнявайки транзакции за добавяне и модифициране на данни и заявки за селекция на съществуващи данни.
- Модул за осигуряване на достъп до документи – позволява интеграция на различни хранилища (файлови, облачни структури и др.), като предоставя съхранение и обработка на физически файлове.

4.3.1.5. Системи за съхранение (Storage systems)

- База данни – централизирано съхранение на структурирани данни.
- Хранилище на файлове – съхранение на техническа и бизнес документация.

4.3.1.6. API интеграция и синхронизация (API Integration and Synchronization Unit)

Този модул осигурява връзка между АСУД и външни системи чрез API, позволявайки интеграция с тикет системи, системи за управление на версиите, платформи за сътрудничество и други инструменти.

4.3.2. Управление на достъпа и потребителите

Ефективното управление на потребителите и контрола върху достъпа е от съществено значение за АСУД. Този компонент гарантира, че информацията в системата е защитена и достъпна само за упълномощени лица, като същевременно осигурява гъвкавост и персонализирани нива на достъп според потребностите на различните потребителски групи.

4.3.2.1. Управление на потребители

Този модул отговаря за регистрацията, идентификацията и управлението на потребителите в системата. Основните му функции включват:

- Регистрация и удостоверяване на самоличност;
- Управление на роли и права на достъп;
- Одитиране и мониторинг на потребителската активност;

4.3.2.2. Управление на достъпа

Този модул гарантира, че потребителите имат достъп само до необходимата им информация, като същевременно защитава чувствителните данни. Основните му функции включват:

- Контрол на достъпа на база роли и разрешения;
- Динамично управление на достъпа въз основа на контекст;
- Модул за заявка и одобрение на достъп;
- Автоматизация на управлението на достъпа чрез ИИ;

4.3.3. Интеграция с ИИ

Интеграционният модул с ИИ позволява конфигуриране на различни външни инструменти с ИИ, съобразени с конкретни профили:

- Интелигентни асистенти;
- Кодификация на знания;
- Откриване на знания;

- Услуги за интеграция с ИИ.

4.4. Разработване на прототип на АСУД

Разработването на прототип на АСУД е стъпка в демонстрирането на приложимостта и ефективността на концептуалната рамка (proof of concept) като се поставя за цел да онагледят как интелигентни агенти и автоматизирани механизми могат да подобрят достъпността, разбирането и управлението на сложна софтуерна документация.

Прототипът позволява валидация на основните функционалности, като управление на документация според различните области на знания, интелигентно търсене и анализ на съдържанието, както и управление на потребителски достъп. Чрез него ще се верифицират интеграциите с интелигентни агенти и възможностите за конфигурация.

4.4.1. Функционални изисквания

Настоящият прототип обхваща различните типове знания, като ги съпоставя с представения по-горе концептуален модел. С цел да се подчертае значението на интелигентните асистенти, функционалностите са дефинирани така, че да осигурят ефективна интеграция с управлението на документи (Таблица 4.3).

Таблица 4.3. Основни функционалности на прототипа

| № | Модул | Основни функционалности |
|---|-------------------------------------|--|
| 1 | Удостоверение за достъп и сигурност | <ol style="list-style-type: none"> 1. Потребителят трябва да може да се аутентикира в системата посредством делегирани потребителско име и парола. 2. Потребителят трябва да може да се оторизира според делегираната му роля. |
| 2 | Мои документи | <ol style="list-style-type: none"> 3. Потребителят трябва да разполага с „Лично пространство“ за съхранение и управление на файлове. 4. Потребителят трябва да може да осъществява качване, сваляне и организиране на документи и папки. 5. Потребителят трябва да може да осъществява анализ на документи чрез интелигентен асистент – генериране на резюмета и класификация на документи. |
| 3 | Процеси | <ol style="list-style-type: none"> 6. Потребителят трябва да може да дефинира оперативни процеси за разработка на софтуер. 7. Потребителят трябва да може да създава процедурни процеси. 8. Потребителят трябва да може да дефинира проектни процеси, по които могат да се стартират нови проекти. |
| 4 | Проекти | <ol style="list-style-type: none"> 9. Потребителят трябва да може да създава проекти с параметри – бюджет и времеви обхват. 10. Потребителят трябва да може да прикачва документи и да управлява проектна документация. 11. Потребителят трябва да може да осъществява качване, сваляне и организиране на документи и папки към даден проект. 12. Потребителят трябва да може да осъществява анализ на документи към проект чрез интелигентен асистент – генериране на резюмета и класификация на документи. |

4.4.2. Нефункционални изисквания и ограничения

Нефункционалните изисквания към прототипа покриват основните изисквания за АСУД. Сред тях ключови аспекти са сигурността, ефективността, интуитивността на интерфейса, както и интеграцията с интелигентен асистент.

Прототипът трябва да гарантира, че само оторизирани потребители имат достъп до чувствителни документи, процеси и проектна информация.

В допълнение трябва да поддържа базови файлови формати (PDF, DOCX, TXT, XLSX, CSV), за да позволи извличане на текстови данни, анализ и интеграция с интелигентен асистент. Това е важно, тъй като различните бизнес процеси включват работа с текстови документи, формуляри и структурирани файлове.

Функционалностите на системата трябва да бъдат представени чрез интуитивен интерфейс, който позволява лесна навигация. Това намалява времето за адаптация на потребителите и увеличава ефективността на работа.

Интеграцията на интелигентен асистент изисква спазването на следните приоритетни условия:

- Интелигентният асистент трябва да бъде интегриран чрез API, осигурявайки стандартизирана комуникация между прототипа и външния модел с ИИ.
- Системата трябва да изпраща и получава информация в структуриран формат (JSON/XML) с възможност за допълнителна валидация и обработка на резултатите.

4.4.3. Дизайн и софтуерна архитектура на прототип на АСУД

Прототипът на АСУД е проектиран като мащабируемо и сигурно уеб-базирано решение, което улеснява управлението на документи, процеси и проекти в организацията. Архитектурата ѝ следва многослоен подход, като основните компоненти включват клиентска част (Client), сървърна част (API) и база данни.

4.4.3.1. Сървърна част

Сървърната част е реализирана като .NET API компонента, която осигурява бизнес логиката и управлението на данните. Той е структуриран в няколко слоя:

4.4.3.1.1. API слой

Системата предоставя RESTful API точки за достъп, които позволяват изпълнението на операции, свързани с управлението на потребители, роли, процеси, проекти и свързаните с тях документи. На това ниво се реализират механизмите за автентикация и оторизация.

При заявка за достъп, потребителят се идентифицира чрез потребителско име и парола. След успешна проверка на валидността на въведените данни, системата генерира токен, съдържащ информация за ролята на потребителя. На база тази роля се определят нивата на достъп и разрешените операции:

- Администратор (Admin) – има пълни права за управление на потребители, процеси и проекти. Може да преглежда, редактира и управлява данните на всички потребители, включително техните роли.

- Потребител (User) – има достъп до личното си файлово пространство, създадените от него процеси и проекти, както и до публично достъпните процеси в системата.

4.4.3.1.2. Слой на услугите

Слоят, съдържащ бизнес логиката, определя начина на изпълнение на операциите и валидацията на данните в системата. Той предоставя функционалности за създаване, прочитане, редактиране и изтриване на потребители, роли, процеси, проекти и документи. Слоят включва операции за управление на дървовидни структури, необходими за организирането на документите в персонализирани пространства и проектни хранилища.

4.4.3.1.3. Слой на областите

Този слой включва следните основни абстракции:

- **Identity** – същности, чиито идентификатори се съхраняват под формата на GUID, осигурявайки уникалност и независимост от релационни ограничения.
- **Entity** – същности, които съдържат одитни данни (дата на създаване, дата на последна промяна и автор на промените).
- **Stage** – същности, които представляват стъпки в даден процес и могат да бъдат номерирани за определяне на реда на изпълнение.
- **Administrator** – същности, които имат администратор (например процеси и проекти), определящи правата за управление.
- **Document** – същности, които представят документи от различни типове – потребителски, процесни и проектни.
- **DocumentTreeNode** – същности, които изграждат дървовидна структура от документи, осигурявайки йерархична организация.

4.4.3.2. Бази данни

За управление на данните се използва Microsoft SQL Server (MSSQL). Базата данни е проектирана да поддържа:

- Йерархични структури за организация на документи и процеси;
- Свързани таблици за съхранение на потребители, роли, права за достъп и проектни параметри;
- Индексирана структура за бързо търсене и извличане на данни;

На базата на предоставената схема на базата данни можем да разпределим таблиците към съответните области на знания – поведенчески, технократски, процесни и продуктови. Важен аспект тук е връзката между процес и продукт, която в този случай е представена чрез проектите.

4.4.3.3. Клиентска част

Клиентската част представлява динамично уеб приложение (Single Page Application - SPA), реализиран с Angular. Той осигурява динамичен и удобен интерфейс, който позволява достъп до основните модули на системата. Всяко действие на потребителя се изпраща към сървърната част чрез RESTful API. Достъпът до различните модули зависи от ролята на потребителя, а сесиите се управляват чрез JSON Web Tokens (JWT) за по-

добра сигурност. Клиентската част е разделена на няколко модула – „Администрация“, „Мои документи“, „Процеси“ и „Проекти“

4.4.4. Реализиране на прототипа

4.4.4.1. Основни процеси в прототипа

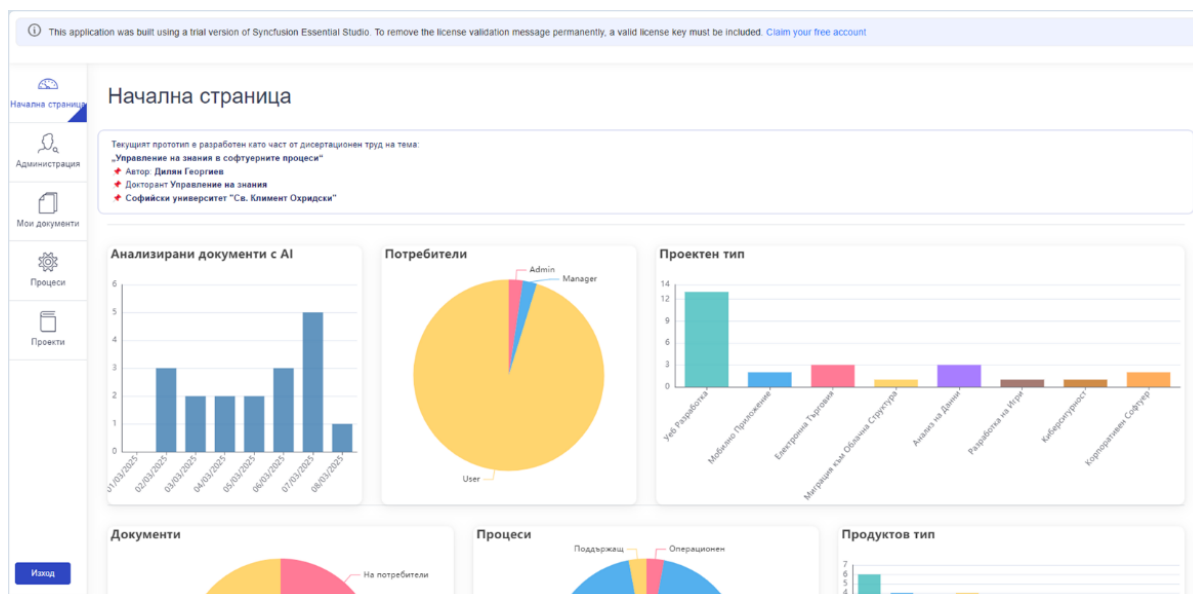
Прототипът обхваща ограничени функционалности на АСУД, таргетирайки областите на знания, от които зависи софтуерният процес и осигурява структурирано управление на документи, потребители, процеси и проекти:

- Начална страница – входна точка за потребителите, откъдето те навигират към различните модули на системата.
- Управление на потребители – включва функционалности за преглед на потребители и създаване на нов потребител, което подсказва наличие на роли или административен контрол.
- Управление на документи на потребител – позволява създаване на папки и качване на документи, което подсказва за организирано съхранение и управление на файлове.
- Управление на процеси – съдържа преглед на процеси и създаване на процес, което показва способността на системата да поддържа и управлява работни потоци и процедури.
- Управление на проекти – включва преглед на проекти и създаване на проект, което предполага, че АСУД има функционалности за проектно-ориентирано управление на документи и процеси.

4.4.4.2. Потребителски интерфейс

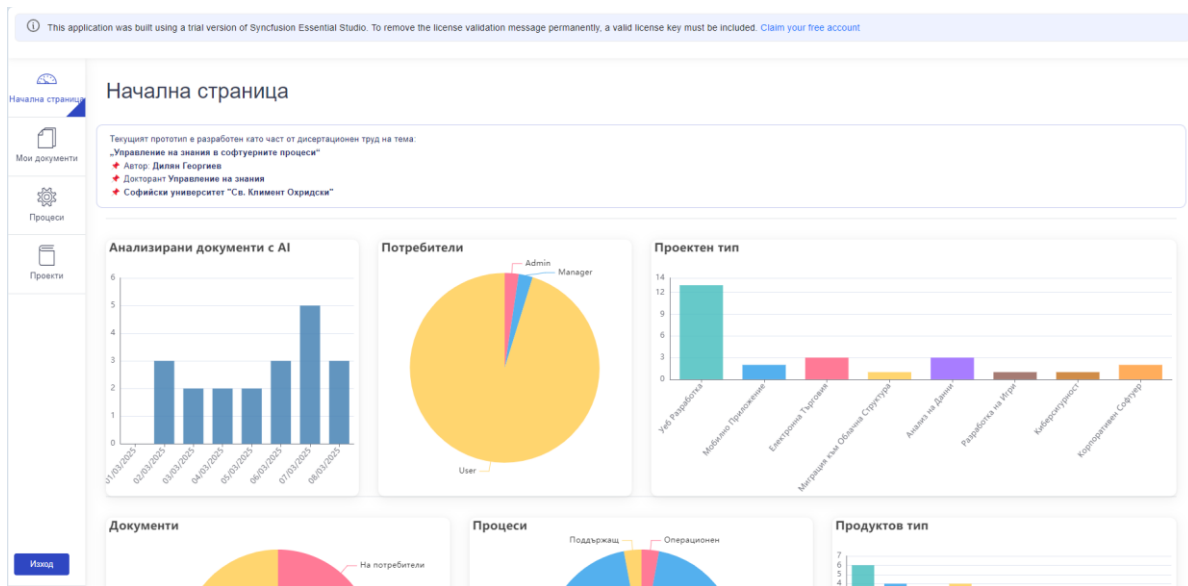
Потребителският интерфейс следва описаните модули на прототипа, дефинирайки два основни изгледа според потребителската роля.

- Администраторски изглед



Фигура 4.4. Изглед на администратор

- Изглед на обикновен потребител



Фигура 4.4. Изглед на обикновен потребител

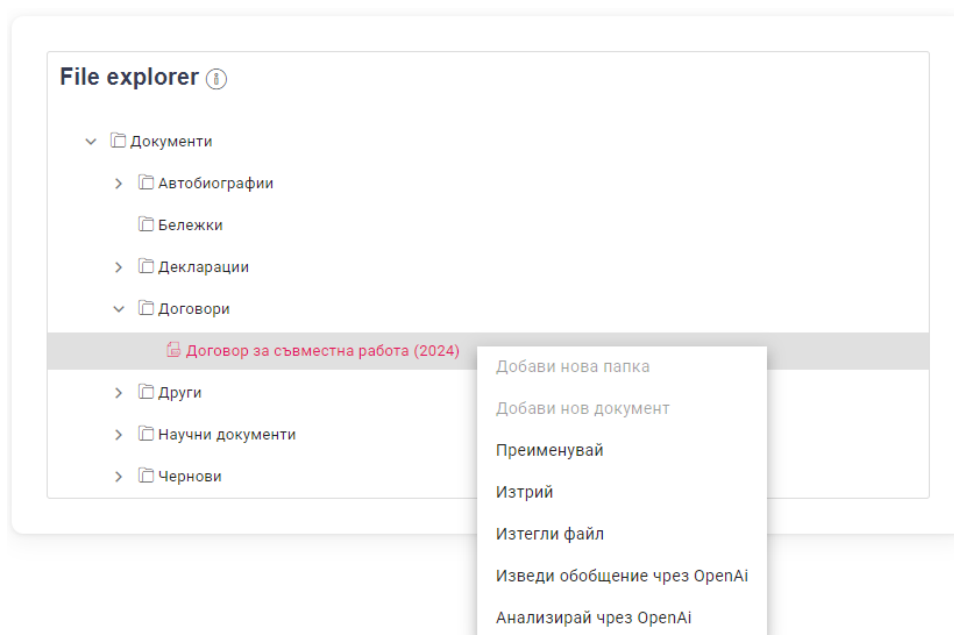
- **Дијаграми:**

Кръгови диаграми са използвани за представянето на текущото състояние на системата: брой качени документи в системата, брой потребители според ролята им, брой създадени процеси според типа им.

Стълбовидни диаграми са използвани за представянето на текущото състояние на системата: брой анализирани документи с ИИ за последните 7 дни, разпределение на създадените проекти и продукти според техния тип.

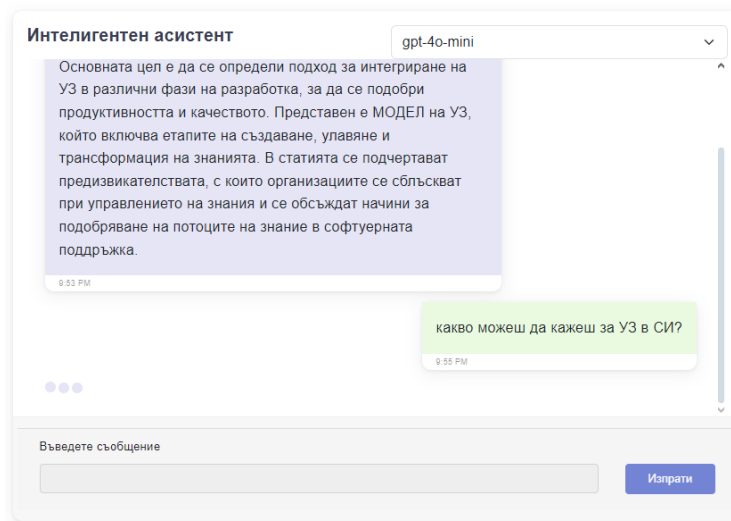
- **Файлово дърво:**

Файловото дърво е основен компонент за управление на документи. Позволява както създаването на папки, така и за качване, изтегляне, изтриване на документи, както и интеракция с интелигентен асистент за обобщение и анализ (Фигура 4.5)



Фигура 4.6: Изглед на файлово дърво за управление на документи

- Интелигентен асистент:



Фигура 4.7: Изглед на компонентата за интелигентен асистент

4.4.4.3. Интеграция с интелигентен асистент

Прототипът на АСУД осъществява интеграция с интелигентен асистент чрез защитена комуникация, иницирана от потребителя. За да се активира връзката с асистента, потребителят предоставя персонализиран API ключ, генериран от своя OpenAI профил. Този ключ се използва за инициализация на HTTP-базирана компонента в системата, която изпълнява ролята на посредник между потребителския интерфейс и OpenAI API.

Потребителят избира документ от интерфейса на системата. След това, съответният файл се извлича от базата данни чрез асинхронна заявка към сървърната част. След получаване на отговора от интелигентния асистент, резултатът се визуализира в чат компонентата на страницата, осигурявайки интуитивна обратна връзка към потребителя.

Успоредно с това, в базата данни се маркира, че файлът е бил обработен от интелигентен агент, като се актуализира състоянието му. По този начин се поддържа прозрачност в историята на обработките и се улеснява проследимостта на действията, извършени чрез ИИ.

4.4.5. Внедряване на прототипа в реална среда

Внедряването на прототипа на АСУД се осъществява в облачната платформа Microsoft Azure, като за целта се създават и конфигурират следните основни ресурси:

- Сървърна част – adms-api;
- Клиентска част – adms-client;
- База данни – adms-db;

4.5. Изводи

Интеграцията на ИИ се превръща в задължителен елемент от всяка система за управление на знания, която поддържа изпълнението на софтуерните процеси. В контекста на управлението на документация АСУД би улеснила достъпа до различните

видове документация, интегрирайки интелигентни агенти във функционалностите си. В тази връзка бяха постигнати няколко ключови резултата. Първо, беше изготвена концептуална рамка за изграждането на такава система, очертаваща нейната роля и значение за оптимизацията на процесите, свързани с управление на знания. Второ, проведено бе проучване на функционалните и нефункционалните изисквания, което даде ясна представа за очакванията и нуждите на потребителите. Трето, на базата на анализа беше предложен архитектурен модел, който интегрира технологии, подпомагащи управлението на знания чрез ИИ. Накрая, беше разработен прототип на АСУД с ограничена функционалност, който демонстрира възможностите на концептуалната рамка в реални условия.

Постигнатите резултати представляват солидна основа за следващата стъпка в изследването – валидация на системата чрез експерименти. В предстоящата глава ще бъде представен процесът на валидация, включващ тестване на разработения прототип, анализ на ефективността му и събиране на обратна връзка от потребителите. Чрез този експериментален подход ще бъде оценена практическата приложимост на предложените решения, като се идентифицират потенциални подобрения и възможности за разширяване на функционалността на АСУД.

5. Валидация на прототип на АСУД

5.1. Методология на валидацията

Експерименталният подход в социалните науки се основава на систематично манипулиране на независими променливи и наблюдение на ефектите им върху зависими променливи (Babbie, 2020). В текущия процес на валидация експериментът ще бъде приложен към тестването на прототип на АСУД, за да се оцени ефективността, потребителското изживяване и възможните подобрения.

5.1.1. Изготвяне на тестови сценарии

Тестовите сценарии са съставени на база основните функционалности на АСУД и са приложени към настоящия труд като приложения. Те са оформени като текстови документи, включващи подробно описание на всяка стъпка, необходима за валидирането на системата. Всеки сценарий проследява ключови действия на потребителите и оценява дали реалните резултати съответстват на очакваните.

- Основна информация за тестовия сценарий;
- Стъпки за изпълнение;
- Резултати от теста;
- Допълнителна информация
- Приложения.

В процеса на тестване на АСУД са приложени пет различни тестови сценария, които обхващат основните модули на системата и нейната интеграция с интелигентния асистент.

- Качване и анализ на документи в модул „Мои документи“;
- Дефиниране на процеси в модул „Процеси“;
- Създаване и управление на проекти в модул „Проекти“;
- Прикачване и анализ на документи в модул „Проекти“;
- Оценка на потребителския интерфейс и интуитивност на системата.

5.1.2. Подготовка на експеримент

Експериментът се провежда чрез контролирано изпълнение на тестовите сценарии от експерти в областта, които изпълняват определени задачи в прототипа на АСУД. Процесът е структуриран в две основни части:

- Изпълнение на тестовия сценарий;
- Попълване на анкета;
- Попълване на тестови формуляри.

5.2. Анализ на резултатите

В анкетното проучване за тестване на прототипа участваха общо 26 специалисти от различни професионални области, свързани със софтуерната индустрия.

Анкетата беше попълнена от разнообразен набор от професионалисти, включително софтуерни разработчици, специалисти по качеството, ръководители на екипи и проекти, бизнес анализатори, DevOps инженери и научни изследователи.

Най-голям дял от респондентите (70%) са софтуерни разработчици, което добавя допълнителна експертиза от гледна точка на ефективност, производителност и дизайн. Включването на проектни мениджъри (23%), ръководители на екипи (17%), DevOps специалисти (10%) и специалисти по качеството (7%) допринася за многостранна оценка на приложимостта на АСУД в областите на знания.

Респондентите са с различни нива на професионален опит, като най-голям дял са специалисти с между 5 и 10 години. Те съставляват 50% от участниците, което предполага, че обратната връзка е базирана на задълбочени практически знания и опит в управлението на документи, процеси и проекти.

Анкетираните са специалисти от различни по мащаб организации, което дава възможност за анализ на приложимостта на системата в различни работни среди.

Най-голям процент от участниците работят в компании със 100 до 500 служители (40%), което е показателно за средно големи организации с добре изградени процеси. Включени са също респонденти от малки екипи с под 20 души, както и от големи предприятия с над 500 служители, което предоставя разнообразни гледни точки за използваемостта на АСУД в различни корпоративни структури.

В хода на тестването респондентите оценяват различните функционалности на АСУД по степен на полезност. Резултатите показват, че най-висока стойност за потребителите има интелигентният асистент за анализ на документи (83%). Това подчертава значимостта на автоматизирания анализ и обобщение на документи като ключова функционалност на системата.

Други добре оценени модули включват:

- Интелигентен асистент към проекти (70%) – потребителите виждат стойност в подпомагането на управлението на документи в рамките на проектите.
- Управление на документи по проект (57%) – подчертава важността на ефективната организация на документи в контекста на проектните дейности.
- Създаване на проекти (53%) – функционалността се оценява като значима за управлението на работния процес.
- Качване на формуляри и Дефиниране на процеси (37% и 40%) – докато тези функции са полезни, резултатите показват, че може да са по-специфични за определени потребителски групи.

По отношение на откритите неизправности се наблюдава, че 37% от участниците не са срещнали никакви проблеми, което е положителен индикатор за стабилността на системата.

Резултатите от проведеното проучване показват висока степен на приемане и положителна нагласа към внедряването на системата. 77% от респондентите заявяват, че биха препоръчали подобна система в своята организация, което свидетелства за нейната полезност и потенциално въздействие върху работните процеси.

Респондентите идентифицират ключови функционалности, които биха допринесли за подобряване на системата и нейната ефективност в реална работна среда:

- Автоматично генериране на отчети (73%);

- Интеграция с външни системи (70%);
- Автоматично категоризиране на документи (57%);
- Контрол на достъпа и права (63%);
- Разширени възможности за търсене (50%);
- Интегриран чат за сътрудничество (47%);
- Управление на версии (23%).

Тези резултати демонстрират, че внедряването на системата е възприето като положителна стъпка към оптимизация на процесите, макар че някои потребители може да имат резерви относно нейното дългосрочно въздействие.

Числовите данни са анализирани чрез прилагане на t-тестове, които еднозначно отхвърлят нулевата хипотеза $H_0: \mu=3$ и приемат алтернативната хипотеза $H_1: \mu>3$ (Таблица 5.1)

Таблица 5.1. Статистическа оценка на прототипа (t-тестове)

| Въпрос | Средна стойност | t-стойност | p-стойност | Резултат |
|--|-----------------|-------------|------------|----------|
| Как оценявате лекотата на използване на системата? | 3.66666667 | 6.020797289 | 7.5224E-07 | $\mu>3$ |
| Доколко системата е интуитивна за навигация и използване? | 3.86666667 | 5.06553905 | 1.0571E-05 | $\mu>3$ |
| Беше ли лесно да извлечете необходимите знания от прикачените документи? | 4.23333333 | 7.869945199 | 5.5734E-09 | $\mu>3$ |
| До каква степен интелигентният асистент би подобрил управлението на знания във ИТ фирма? | 3.56666667 | 3.084323807 | 0.00222541 | $\mu>3$ |

Обратната връзка показва, че АСУД успешно изпълнява основните си функции – анализ на документи, управление на процеси и интеграция с интелигентния асистент. Потребителите оценяват ясно извлечената информация, липсата на халюцинации и интуитивния UX/UI интерфейс.

Въпреки положителните отзиви, са идентифицирани проблеми като забавяне при анализ на файлове, некоректно запазване на дати в модул „Проекти“ и ограничения при качване на документи след използване на OpenAI.

Препоръчва се подобряване на стабилността на интелигентния асистент, отстраняване на бъгове и допълнителни тестове за по-надеждна работа и по-добро потребителско изживяване.

5.3. Изводи

Процесът на валидация на прототипа на АСУД се състои от изпълнението на редица тестови сценарии и извличането на обратна връзка от експерти в областта. За целта беше приложен експериментален подход за оценка на ефективността и приложимостта на АСУД.

В рамките на валидационния процес бяха подбрани професионалисти в СИ, които изпълниха предварително дефинираните тестови сценарии. Освен това те попълниха

анкетен формуляр, който позволи количествена и качествена оценка на прототипа, а някои от тях предоставиха и допълнителна обратна връзка относно възможностите за усъвършенстване на системата. Анализът на получените резултати разкри ключови изводи, свързани с ролята на интелигентния асистент в системите за управление на документация, като се отчете неговият потенциал за оптимизация на процесите, повишаване на ефективността и подобряване на потребителското изживяване.

В заключение, проведеното изследване демонстрира, че прототипът на АСУД е качествено валидиран, като представените данни доказват стойността и приложимостта на предложената концептуална рамка. Резултатите от експерименталния анализ потвърждават, че разработената система отговаря на предварително дефинираните функционални и нефункционални изисквания, а интегрирането на изкуствен интелект в процеса на управление на документация носи значителни предимства. Тези заключения оформят основата на финалната част на дисертацията, където ще бъдат обобщени основните научни и практически приноси на настоящото изследване, както и насоки за бъдещо развитие и усъвършенстване на концептуалната рамка.

6. Заключение

6.1. Ползи от предложената АСУД

В съвременните ИТ компании управлението на документи преминава от статични файлови хранилища към динамични, интегрирани платформи, които използват изкуствен интелект за анализ и препоръки.

АСУД има значително влияние върху ефективността на различните специалисти в софтуерната индустрия, като подпомага съхранението, анализирането и споделянето на информация. Всеки тип експерт извлича специфични ползи от системата, които допринасят за повишаване на производителността и качеството на работа.

АСУД трансформира проектния мениджмънт, като подпомага различните роли в екипа и оптимизира потока на документи. Мениджърите на проекти получават бърз достъп до ключова информация чрез интелигентно търсене и обобщение, което ускорява взимането на решения. Бизнес анализаторите могат да идентифицират критични тенденции и зависимости, докато разработчиците и техническите екипи имат достъп до актуални спецификации и кодова документация.

АСУД е стратегически инструмент, който улеснява управлението на знания и подобрява комуникацията между различните роли в софтуерната индустрия. Чрез автоматизиране на процесите, интегриране на документация и осигуряване на лесен достъп до информация, системата допринася за по-добро качество, по-ефективна работа и повишена конкурентоспособност на организациите.

Като обобщение може да се твърди, че текущата концептуална рамка е основен принос на дисертацията, като се потвърждава основната хипотеза на изследването, че УЗ е важен компонент от софтуерния процес, който може да бъде значително подпомогнат чрез интеграцията на системи, използващи инструменти, базирани на изкуствен интелект.

6.2. Постигнати резултати и бъдещи възможности

Настоящият дисертационен труд е посветен на изследването, моделирането и приложението на изкуствения интелект в управлението на знания в софтуерните процеси, като в него са изпълнени последователно редица ключови задачи. В началото е проучено текущото състояние на УЗ в СИ, което включва анализ на областите на знания в СИ, както и идентифициране на технологиите, подпомагащи УЗ. На тази основа е разработен модел на областите на знания в СИ, който систематизира основните категории знания, необходими за ефективното управление на софтуерните процеси.

Резултатите от дисертационния труд откриват широк спектър от възможности за развитие както на АСУД, така и на приложенията на изкуствения интелект в управлението на знания. Една от основните посоки е разширяване на интелигентните функционалности, включително автоматизирано обобщаване и анализ на документи, персонализирани препоръки на знания и адаптивно управление на информационните потоци в софтуерните процеси.

В заключение с развитието на големите езикови модели и генериращия изкуствен интелект се открива възможността за динамично и проактивно управление на знания,

при което системата самостоятелно анализира тенденции, предсказва бъдещи нужди и автоматично генерира препоръки. Така, резултатите от настоящото изследване могат да послужат като основа за създаването на още по-интелигентни и адаптивни системи, които да подпомагат индустриалната практика и научните изследвания в областта на софтуерното инженерство.

Основни приноси на дисертационния труд

Научно-приложни приноси

1. Направен е анализ и класификация на основните области на знания в софтуерното инженерство (Глава 1).
2. Разработен е модел на областите на знания в софтуерното инженерство (Глава 2).
3. Разработена е концептуална рамка на Автоматизирана система за управление на документация, интегрираща инструменти базирани на изкуствен интелект (Глава 3).

Приложни приноси

1. Проучени са и анализирани проблемите в софтуерното инженерство, които засягат областите на знания (Глава 2).
2. Разработен е и валидиран прототип на автоматизирана система за управление на документи (Глави 3 и 4).

Публикации по дисертацията

1. Georgiev, D. (2024). Exploring software engineering knowledge domains. *Ann. Sofia Univ. Fac. Math. And Inf.*, 111, 35–53. <https://doi.org/10.60063/gsu.fmi.111.35-53> (Scopus)
2. Georgiev, D., and Antonova, A. (2024). Enhancing Knowledge Sharing Processes via Automated Software Documentation Management Systems Using Gen AI Software Tools, *2024 XXXIV International Scientific Symposium Metrology and Metrology Assurance (MMA)*, Sozopol, Bulgaria, 2024, pp. 1-6, doi: 10.1109/MMA62616.2024.10817681 (Scopus)
3. Georgiev, D. (2023) Exploring knowledge management from a software engineering perspective, *European Conference on Knowledge Management*, Vol. 24 No. 2 (2023): *Proceedings of the 24th European Conference on Knowledge Management*, pp. 1571-1578 DOI: 10.34190/eckm.24.2.1497 (Scopus, SJR)

цитиране в Scopus: Alkhatib, G. (2024). Verification and Validation of Knowledge Engineering Systems: A Life Cycle Framework. In: Mirzazadeh, A., Molamohamadi, Z., Erdebilli, B., Babae Tirkolae, E., Weber, GW. (eds) *Science, Engineering Management and Information Technology. SEMIT 2023. Communications in Computer and Information Science*, vol 2198. Springer, Cham. https://doi.org/10.1007/978-3-031-72284-4_3

цитиране в Scopus: Darwish, H., Darwish, W., Darwish, H., AlHmoud, I. W., & Alshraideh, O. Artificial Intelligence, Machine Learning, and Deep Learning Applications in the Engineering Fields—A Comprehensive Review. *Sigma J Eng Nat Sci*, DOI: 10.14744/sigma.2024.00126

Референции

- Ahmad, F. & Karim, M. (2019). *Impacts of knowledge sharing: a review and directions for future research*, Journal of Workplace Learning, 31(3), 207-230. DOI/10.1108/JWL-07-2018-0096, Available at SSRN: <https://ssrn.com/abstract=3392007>
- Alzahrani, A. (2024). *Software Systems Documentation: A Systematic Review*. International Journal of Advanced Computer Science and Applications, 15(8). <https://doi.org/10.14569/ijacsa.2024.0150816>
- Arisha, A., Ragab, M. (2013). *Knowledge Management and Measurement: A Critical Review*. Journal of Knowledge Management, Vol. 17 (2013) Issue: 6, pp 873-901. <https://doi.org/10.1108/JKM-12-2012-0381>
- Aversano, L., Guardabascio, D., Tortorella, M. (2017). *Analysis of the Documentation of ERP Software Projects*. 121, 423–430. <https://doi.org/10.1016/J.PROCS.2017.11.057>
- Ayarza, V.C. & Bayona-Oré, S. (2020). *Cluster Monitoring and Integration in Technology Company*. In: Mejia, J., Muñoz, M., Rocha, Á., A. Calvo-Manzano, J. (eds) Trends and Applications in Software Engineering. CIMPS 2019. Advances in Intelligent Systems and Computing, vol 1071. Springer, Cham. https://doi.org/10.1007/978-3-030-33547-2_19 pp 253–265
- Ayaz A. & Yanartaş M. (2020). *An analysis on the unified theory of acceptance and use of technology theory (UTAUT): Acceptance of electronic document management system (EDMS)*. Computers in Human Behaviour Reports, Volume 2, 2020, 100032, ISSN 2451-9588, <https://doi.org/10.1016/j.chbr.2020.100032>
- Babbie, E. (2020). *The practice of social research (15th ed.)*. CENGAGE Learning Custom Publishing.
- Baydoun, Ali & Ahmad, Fatima (2021). *Knowledge base cloud framework for effective knowledge retention in software supply chain*. International Journal of Engineering Applied Sciences and Technology. 6. 44-52. 10.33564/IJEAST.2021.v06i08.007
- Bedjeti, A., Lago, P., Lewis, G. A., De Boer, R. D., Hilliard, R. (2017). *Modeling Context with an Architecture Viewpoint*. 117–120. <https://doi.org/10.1109/ICSA.2017.26>
- Borges A., Laurindo F., Spínola M., Gonçalves R., Mattos C. (2021). *The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions*, International Journal of Information Management, Volume 57, 2021, 102225, ISSN 0268-4012, <https://doi.org/10.1016/j.ijinfomgt.2020.102225>
- Broy, M. (2022). *Software System Documentation: Coherent Description of Software System Properties* (pp. 10–27). https://doi.org/10.1007/978-3-031-19756-7_2
- Cooperman, M. S., & Karch, R. (1997). *Hierarchically arranged knowledge domains*. <https://patents.google.com/patent/US6247007B1/en>
- Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative and Mixed Methods Approaches (4th ed.)*. Thousand Oaks, CA: Sage.

- Ding, W., Liang, P., Tang, A., & Vliet, H. (2014). *Knowledge-based approaches in software documentation: A systematic literature review*. *Inf. Softw. Technol.*, 56, 545-567. <https://doi.org/10.1016/j.infsof.2014.01.008>
- Georgiev, D. (2024). *Exploring software engineering knowledge domains*. *Ann. Sofia Univ. Fac. Math. And Inf.*, 111, 35–53. <https://doi.org/10.60063/gsu.fmi.111.35-53>
- Georgiev, D., and Antonova, A. (2024). *Enhancing Knowledge Sharing Processes via Automated Software Documentation Management Systems Using Gen AI Software Tools*, 2024 XXXIV International Scientific Symposium Metrology and Metrology Assurance (MMA), Sozopol, Bulgaria, 2024, pp. 1-6, doi: 10.1109/MMA62616.2024.10817681
- Georgiev, D. (2023). *Exploring knowledge management from a software engineering perspective*, European Conference on Knowledge Management, 2023, Volume 2: pp 1571-1578, 10.34190/eckm.24.2.1497
- Gourova, E. (2010). *Knowledge management strategy for Small and Medium Enterprises*. In *Proceedings of the International Conference on Applied Computer Science* (pp. 639-648).
- Gourova, E. & Dragomirova, M. (2015). *Design of knowledge management info-structures*. *Proceedings of the 20th European Conference on Pattern Languages of Programs* (2015)
- Gourova, E. & Toteva, K. (2014). *Design of Knowledge Management Systems*. VikingPLOP 2014.
- Haq, M.A. & Anwar, S. (2016). *A systematic review of knowledge management and knowledge sharing: Trends, issues, and challenges*. *Cogent Business & Management*, 3:1, <https://doi.org/10.1080/23311975.2015.1127744>
- Heisig, P. (2024). *Knowledge management*. 229–255. <https://doi.org/10.4337/9781035343706.00022>
- Huang, S.M., Chu, Y.T, Li, S.H., Yen, D.C. (2008). *Enhancing conflict detecting mechanism for Web Services composition: A business process flow model transformation approach*. In: *Information and Software Technology*, Volume 50, Issue 11, 2008, ISSN 0950-5849, <https://doi.org/10.1016/j.infsof.2007.10.014 pp 1069-1087>
- Hung, W., & Wang, W. (2020). *Design Principles of Wiki System for Knowledge Transfer and Sharing in Organizational Education and Training*. *Sustainability*, 12(17), 6771. <https://doi.org/10.3390/su12176771>
- Jarrahi M.H., Askay D., Eshraghi A., Smith P. (2023). *Artificial intelligence and knowledge management: A partnership between human and AI*, *Business Horizons*, Volume 66, Issue 1, 2023, Pages 87-99, ISSN 0007-6813, <https://doi.org/10.1016/j.bushor.2022.03.002>
- Kirschen M. (1993). *Online documentation: evolution rather than revolution*. In *Proceedings of the 11th annual international conference on Systems documentation (SIGDOC '93)*. Association for Computing Machinery, New York, NY, USA, 177–183. <https://doi.org/10.1145/166025.166080>
- Königstorfer F. & Thalmann S. (2022). *AI Documentation: A path to accountability*, *Journal of Responsible Technology*, Volume 11, 2022, 100043, ISSN 2666-6596, <https://doi.org/10.1016/j.jrt.2022.100043>

- Lim, S.L.O., Lim, H.M., Tan, E.K., Tan, TP. (2020). *Examining Machine Learning Techniques in Business News Headline Sentiment Analysis*. In: Alfred, R., Lim, Y., Havaluddin, H., On, C. (eds) Computational Science and Technology. Lecture Notes in Electrical Engineering, vol 603. Springer, Singapore, https://doi.org/10.1007/978-981-15-0058-9_35
- Machuca-Villegas, L., Gasca-Hurtado, G.P. (2020). *Towards a Social and Human Factor Classification Related to Productivity in Software Development Teams*. In: Mejia, J., Muñoz, M., Rocha, Á., A. Calvo-Manzano, J. (eds) Trends and Applications in Software Engineering. CIMPS 2019. Advances in Intelligent Systems and Computing, vol 1071. Springer (2020), Cham. https://doi.org/10.1007/978-3-030-33547-2_4 pp 36–50
- Mahmood, A., Okumus, I. (2017). *Design and Implementation of an Electronic Document Management System*. Mehmet Akif Ersoy Üniversitesi Uygulamalı Bilimler Dergisi, 1(1), 9-17. Year 2017 Volume: 1 Issue: 1 <https://doi.org/10.31200/makuubd.321093>
- Marinho, M., Luna, A., Beecham, S. (2018). *Global Software Development: Practices for Cultural Differences*. In: , et al. Product-Focused Software Process Improvement. PROFES 2018. Lecture Notes in Computer Science, vol 11271. Springer, Cham. https://doi.org/10.1007/978-3-030-03673-7_22 pp 299–317
- Moghaddam, R., Garg, S., Clement, C., Mohylevskyy, Y., Sundaresan, N. (2022). *Generating Examples from CLI Usage: Can Transformers Help?* Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. <https://doi.org/10.1145/3534678.3549983>
- Navidi, F., Hassanzadeh, M., Zolghadr Shojai, A. (2019). *Organizational knowledge documentation in project-based institutes*. <https://www.emerald.com/insight/content/doi/10.1108/EL-10-2015-0196/full/html>
- Ogayar-Anguita, C. J., López-Ruiz, A., Segura-Sánchez, R. J., & Rueda-Ruiz, A. J. (2023). *A Version Control System for Point Clouds*. Remote Sensing, 15(18), 4635. <https://doi.org/10.3390/rs15184635>
- Osorio Angel, S., Peña Pérez Negrón, A., Espinoza Valdez, A. (2019) From a Conceptual to a Computational Model of Cognitive Emotional Process for Engineering Students. In: Mejia, J., Muñoz, M., Rocha, Á., A. Calvo-Manzano, J. (eds) Trends and Applications in Software Engineering. CIMPS 2019. Advances in Intelligent Systems and Computing, vol 1071. Springer (2019), Cham. https://doi.org/10.1007/978-3-030-33547-2_14 pp 173–186
- Parnas, D. L. (2021). *Software Engineering: A Profession in Waiting*. IEEE Computer, 54(05), 62–64. <https://doi.org/10.1109/MC.2021.3057685>
- Polyakov, M., Khanin, I., Bilozubenko, V., Korneyev, M., & Nebaba, N. (2020). *Information technologies for developing a company's knowledge management system*. 4(1), 15–25. [https://doi.org/10.21511/KPM.04\(1\).2020.02](https://doi.org/10.21511/KPM.04(1).2020.02)
- Raglianti, M., Nagy, C., Minelli, R., Lin, B., & Lanza, M. (2023). *On the Rise of Modern Software Documentation (Pearl/Brave New Idea)*. In: 37th European Conference on Object-Oriented Programming (ECOOP 2023). Leibniz International Proceedings in Informatics (LIPIcs), Volume 263, pp. 43:1-43:24, Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2023) <https://doi.org/10.4230/LIPIcs.ECOOP.2023.43>

Regla, A.I. & Marquez, P.S. (2020). *Workplace Document Management System Employing Cloud Computing and Social Technology*. In: Alfred, R., Lim, Y., Havaluddin, H., On, C. (eds) *Computational Science and Technology. Lecture Notes in Electrical Engineering*, vol 603. Springer, Singapore. https://doi.org/10.1007/978-981-15-0058-9_40

Shetty, M., Panchami, B. S., & Gadiyar, H. M. T. (2023). *Software Development Life Cycle (SDLC) in Software Engineering – A Brief Review*. 5–9. <https://doi.org/10.48001/jocsss.2023.115-9>

Torres, V., Gil, M., Pelechano, V. (2019). *Software Knowledge Representation to Understand Software Systems*. In: Franch, X., Männistö, T., Martínez-Fernández, S. (eds) *Product-Focused Software Process Improvement. PROFES 2019. Lecture Notes in Computer Science()*, vol 11915. Springer (2019), Cham. https://doi.org/10.1007/978-3-030-35333-9_10 pp 137–144

Wiig, K. (2004). *People-Focused Knowledge Management*, Butterworth-Heinemann, 2004.

ДЕКЛАРАЦИЯ ЗА ОРИГИНАЛНОСТ

Декларирам, че резултатите и приноси в моята дисертация, „Управление на знания в софтуерните процеси“, подадена за образователна и научна степен “Доктор” в Софийски университет „Св. Климент Охридски“, са оригинални и не произтичат от изследвания, в които не съм участвал. Не са използвани никакви материали, защитени с авторски права, без да бъде посочен техният източник.

Тази дисертация не е подавана за научна степен в друго учебно заведение и цялата представена информация, включително публикации, документи и експериментални резултати, е точна и вярна.

Подпис: