

Taras Shevchenko National University of Kyiv
Software Systems and Technologies Department
Київський національний університет імені Тараса Шевченка
Кафедра програмних систем і технологій

France, LeMann , University LeMann,

Bulgaria, Sofia, University of Library Studies and Information Technologies

**Czech Republic, Brno, University of Technology Institute of Mathematics and
Descriptive Geometry**

MSTIoE 2019-5
5-th East European Conference on
Mathematical Foundations and Software
Technology of Internet of Everything

5-та Східно-Європейська конференція
Математичні та програмні технології Internet
of Everything

(19-20.12.2019, Kyiv - Київ)

Proceedings - Збірник матеріалів

Збірник внесений до наукової електронної бібліотеки
Національної бібліотеки України імені Вернадського

УДК 002.5:004

MSTIoE 2019-5. 5-та Східно-Європейська конференція “Математичні та програмні технології Internet of Everything” (19-20.12.2019, Київ). Зб.матер., КНУ ім.Т.Шевченка, 106с.

У збірнику узагальнені матеріали конференції, яка проходила на базі кафедри програмних систем і технологій (ПСТ) Київського національного університету імені Тараса Шевченка 19-20.12.2019 року. В матеріалах висвітлюються актуальні питання розвитку теорії та практики програмування Internet of Everything (Всеохоплюючий інтернет, IoE) та розвитку суміжних технологій.

Напрямки конференції:

- Математичні технології IoE;
- Програмні технології IoE;
- Апаратні технології IoE;
- Інші технології та види забезпечення IoE.

Збірник розрахований на представників бізнесу та державних органів, викладачів та наукових працівників, аспірантів, студентів які займаються питаннями програмування Інтернет-додатків та розвитку інформаційних технологій в цілому.

MSTIoE 2019-5. 5-th East European Conference on Mathematical Foundations and Software Technology of Internet of Everything. (19-20.12.2019, Київ). Proceedings, Taras Shevchenko National University of Kyiv, 106 pp.

The collection summarizes the materials of the conference, which took place on the basis of the Department of Software Systems and Technologies (PST) of the Taras Shevchenko Kyiv National University on the 19-20.12.2019. The materials cover the topical issues of the development of the theory and practice of programming Internet of Everything (IoE) and the development of related technologies.

Directions of the conference:

- IoE Mathematical Technologies;
- IoE Software Technologies;
- Hardware IoE technology;
- Other technologies and types of IoE security.

The collection is intended for representatives of business and government bodies, teachers and researchers, postgraduates, students who are involved with the issues of programming of Internet applications and the development of information technologies in general.

MSTIoE 2019-5. 5-я Восточно-Европейская конференция "Математические и программные технологии Internet of Everything" (19-20.12.2019, Киев). Сб матер., КНУ им.Т.Шевченко, 106 с.

В сборнике обобщены материалы конференции, которая проходила на базе кафедры программных систем и технологий (ПСТ) Киевского национального университета имени Тараса Шевченко 19-20.12.2019 года. В материалах освещаются актуальные вопросы развития теории и практики программирования Internet of Everything (Всеобъемлющий интернет, IoE) и развитию смежных технологий.

Направления конференции:

- Математические технологии IoE;
- Программные технологии IoE;
- Аппаратные технологии IoE;
- Другие технологии и виды обеспечения IoE.

Сборник рассчитан на представителей бизнеса и государственных органов, преподавателей и научных работников, аспирантов, студентов, которые занимаются вопросами программирования Интернет-приложений и развития информационных технологий в целом.

Редакційна колегія:

Бичков О.С. , д.т.н., доц. (голова),	завідуючий каф. Програмних систем і технологій Київський національний університет імені Тараса Шевченка
Шевченко В.Л. , д.т.н., професор (заступник голови)	професор каф. Програмних систем і технологій Київський національний університет імені Тараса Шевченка
Michele Pizerill	President of LeMann University, LeMann, France
Yannick Estève	Professor of Computer Science Director of the IICC of LeMann University, LeMann, France
Georgy Dimitrov	Vice Dean of University of Library Studies and Information Technologies, Sofia, Bulgaria
Eugenia Kovacheva , PhD	Assoc. Professor of University of Library Studies and Information Technologies, Sofia, Bulgaria
Josef Diblík DrSc., prof. RNDr	Head of department of Mathematics and Descriptive Geometry of University of Technology Institute, Brno, Czech Republic
Хусаїнов Д.Я. , д.ф.м. наук, професор	Київський національний університет імені Тараса Шевченка
Шатирко А.В. , к.ф.м.наук, доцент	Київський національний університет імені Тараса Шевченка
Вишнівський В.В. , д.т.н., професор	Завідуючий кафедри Комп'ютерних наук Державного університету телекомунікацій (ДУТ), м.Київ
Онщенко В.В. , д.т.н., доцент	Завідуюча кафедри Інженерії програмного забезпечення Державного університету телекомунікацій, м.Київ
Курченко О.А. , к.т.н., доцент	доцент кафедри Інформаційної та кібернетичної безпеки Державного університету телекомунікацій, м.Київ
Щебланін Ю.М. , к.т.н., доцент	Доцент кафедри Управління інформаційною та кібернетичною безпекою Державного університету телекомунікацій, м.Київ
Лисенко О.І. , д.т.н., професор	Професор інституту телекомунікацій НТУУ (КПІ)
Чумаченко С.М. , д.т.н., с.н.с.	Завідувач кафедри Інформаційних систем Національного університету харчових технологій
Приставка П.О. , д.т.н., професор	Національний авіаційний університет, завідувач кафедри
Коваленко Л.Б. , к.г.н., доцент	Одеський державний екологічний університет, декан факультету,
Кузнichenko С.Д. , к.г.н., доцент	Одеський державний екологічний університет, заступник декана
Рибидайло А.А. , к.т.н.	провідний науковий співробітник науково-дослідного управління Проблем розвитку інформаційних технологій Центру воєнно-стратегічних досліджень Національного університету оборони України

Матеріали подано в авторській редакції

Відповідальний за випуск: Шевченко В.Л.

Адреса редакції: 03116 Київ-116, вул. Богдана Гаврилишина (В.Василевської), 24
тел. (066) 430-54-32

<http://pst.knu.ua/MSTIoE>

© кафедра ПСТ КНУ ім.Тараса Шевченко, 2019

ЗМІСТ

Козачок П.А., Бражененко М.Г., Шевченко В.Л.	Хмарно-орієнтований дизайн системи систем	15
Журко О. О., Шевченко В. Л.	Фільтр Калмана	16
Петракивський Д.О., Шевченко В.Л.	Кластеризація нерухомості на основі особливостей ціни та розташування	17
Кузьменко Ольга, Поперешняк С.В.	Визначення хмари за допомогою згорткових нейронних мереж	18
Бичков О.С., Берестов Д.С., Шевченко В.В.	Моделі реального життя на основі клітинних автоматів	19
Ільїн М., Василенко В., Шевченко В.Л.	Проект PEERGAZE керування мобільним пристроєм за допомогою погляду	22
Марчук В.В., Шевченко В.Л.	Проект «Temp-sensor»	24
Бичков О.С., Шевченко В.Л., Беляченко О.В.	Спіраль розвитку інженерії програмного забезпечення та менеджменту проектів програмного забезпечення	26
Ляшуга Микола, Ковтун О.І.	GitLab в командній розробці	30
Самойленко Олег, Супрун О.М.,	Bitbucket для командної розробки програмного забезпечення	32
Мірошник Олексій, Шевченко В.Л.	Розподілений інструмент управління вихідним кодом Mercurial	35
Баснакаєв Салават, Берестов Д.С.	Сервіс GitHub	37
Сітко Денис, Юрчук І.А.	Microsoft Project програмне забезпечення для управління проектами	41
Дронь Дмитро, Меркулова К.В.	Worksection - сервіс управління проектами для бізнесу	43
Цехмейструк Андрій, Доценко С.І.	JIRA - засіб відстеження завдань та управління проектами	46
Панасовський Дмитро, Ковалюк Т.В.	Notion - єдиний інструмент для ведення особистих та корпоративних справ і задач	52
Шаламай Єлизавета, Поляков С.А.	Microsoft Teams - уніфікована платформа для спілкування та співпраці	55
Шапенко Євгеній, Юрчук І.А.	Органайзер та поштовий клієнт MS Outlook	59

Рибалочка Михайло, Супрун О.М.	Confluence – програмне забезпечення командної роботи	61
Шестаков Андрій, Поляков С.А.	Програма телекомунікацій Skype	63
Балацкий Дмитро, Меркулова К.В.	Dropbox – програмне забезпечення командної розробки	66
Дяченко Ігор	Evernote - інструмент для командної розробки програмного забезпечення	69
Ткаченко М.В., Федоренко Р.М., Шевченко В.Л.	AcceleratedSAP (ASAP)	72
Берестов Д.С., Курченко О.А., Покотило О.І.	Особливості розробки програмного забезпечення для забезпечення захисту інформації від несанкціонованого доступу	76
Берестов Д.С., Курченко О.А., Покотило О.І.	Побудова комплексної системи захисту інформації в автоматизованих системах	79
Бражиненко Максим, Ляшуга Микола, Самойленко Олег, Дронь Дмитро, Цехмейструк Андрій, Шаламай Єлизавета, Беляченко О.В.	Аналіз відповідності навчальних дисциплін вимогам виробництва програмного забезпечення	82
Нестеренко О.В., Нетесін І.Є., Поліщук В.Б.	Інформаційно-аналітична система територіального управління якістю атмосферного повітря	84
Артеменко Є.В., Мідоян Е.Р.	Обробка даних комп'ютерним зором в інформаційному просторі за допомогою лінійної фільтрації	86
Карандась Д.Д., Ткаченко М.В., Федоренко Р.М.	Використання нейронних мереж при оцінці якості ПЗ	87
Холявкін Іван, Ковалюк Т.В.	Автоматизація визначення переліку дисциплін за спеціальністю «Інженерія програмного забезпечення» на основі мотиваційного листа студента	89
Соколов К.О., Гудима О.П.	Питання забезпечення кібербезпеки у всесвітньому інформаційному середовищі	91
Петрівський В.Я., Шевченко В.Л., Бичков О.С.	Використання 5G технологій у Інтернеті Речей та Інтернеті Всього	93
Ващук І., А. Поперешняк С.В.	Автоматизація обліку та моніторингу робочого часу на підприємстві згідно чинного законодавства	94
Соколенко П.Ю., Поперешняк С.В.	Система обробки даних паркування автомобілів, як частина Smart City	97

Поперешняк С.В.	Деякі аспекти тестування псевдовипадкової послідовності малої довжини	101
Прудченко А.О., Старкова С.В., Герасименко К.В.	Алгоритм співставлення зображень за ключовими точками при масштабуванні і обертанні об'єктів	104
Мирошников С.О., Карнацький Д.О., Бичков О.С.	Інформаційно-аналітична система електронного голосування з використанням протоколу двох агентств Фудзіоки-Окамото-Охта	106

CONTENTS

Kozachok P.A., Brazenenko M.G., Shevchenko V.L.	Cloud based architecture design of system of systems	15
Zhurko O. O., Shevchenko V.L.	Kalman filter	16
Petrakivsky D.O., Shevchenko V.L.	Real estate clustering, based on the price and placement features	17
Kuzmenko Olha, Popereshnyak S.V.	Cloud detection using convolution neural network	18
Bychkov O.S., Berestov D.S., Shevchenko V.V.	Real life models with cellular automaton	19
Ilyin M., Vasilenko V., Shevchenko V.L.	The PEERGAZE project controls your mobile device with a view	22
Marchuk V.V., Shevchenko V.L.	Project «Temp-sensor»	24
Bichkov O.S., Shevchenko V.L., Belyachenko O.V.	The spiral of development of software engineering and software project management	26
Lyashug Mykola, Kovtun O.I.	GitLab in command development	30
Samoilenko Oleg, Suprun O.M.	Bitbucket for team software development	32
Miroshnik Oleksiy, Shevchenko V.L.	Distributed Mercurial Source Control Tool	35
Basnakaev Salavat, Berestov D.S.	GitHub Service	37
Sitko Denis, Yurchuk I.A.	Microsoft Project project management software	41
Dron Dmitry, Merkulova K.V.	Worksection - project management service for business	43
Tsehmeystruk Andriy, Dotsenko S.I.	JIRA is a tool for task tracking and project management	46
Dmitry Panasovsky, Kovaluk T.V.	Notion is the unified tool for personal and corporate affairs and tasks	52
Shalamay Elizaveta, Polyakov S.A.	Microsoft Teams is a unified platform for communication and collaboration	55
Shapenko Eugene, Yurchuk I.A.	MS Outlook Organizer and Email Client	59

Rybalochka Michaylo, Suprun O.M.	Confluence - teamwork software	61
Shestakov Andrey, Polyakov S.A.	Skype telecommunication program	63
Balatsky Dmitry, Merkulova K.V.	Dropbox is a team development software	66
Dyachenko Igor	Evernote is a tool for team development of software	69
Tkachenko M.V., Fedorenko R.M., Shevchenko V.L.	AcceleratedSAP (ASAP)	72
Berestov D.S., Kurchenko O.A., Pokotilo O.I.	Software development features to protect information from unauthorized access	76
Berestov D.S., Kurchenko O.A., Pokotilo O.I.	Construction of a complex system of information protection in automated systems	79
Brazhinenko Maxim, Lyashuga Mykola, Samoilenko Oleg, Dron Dmitry, Tsekhmeystruk Andriy, Shalamay Elizaveta, and Belyachenko O.V.	Analysis of compliance of educational disciplines with the requirements of software production	82
Nesterenko O.V., Netesin I.E., Polischuk V.B.	Information-analytical system for atmospheric air quality territorial management	84
Artemenko E.V., Midoian E.R.	Computer vision processing in the information space using linear filtering	86
Karandas D.D., Tkachenko M.V., Fedorenko R.M.	Use of neural networks in software quality assessment	87
Kholiavkin Ivan, Kovaliuk T.V.	Automation of determining the list of disciplines in the specialty "Software Engineering" based on the student's motivation letter	89
Sokolov K. , Hudyma O.	Issues of cyber security in the world information environment	91
Petrivskyi V., Shevchenko V., Bychkov O.	Promising use cases 5g technologies in Internet of Things and Internet of Everything	93
Vashchuk I.A., Popeshnyak S.V.	Automation of accounting and monitoring of working time at the enterprise according to the current legislation	94
Sokolenko P.Yu., Popeshnyak S.V.	Car parking data processing system as part of Smart City	97

Popereshnyak S.V.	Some aspects of testing a short-length pseudorandom sequence	101
Prudchenko A.A., Starkova S.V., Herasymenko K.V.	Algorithm of comparing images of scaled and rotated objects based on key points	104
Myroshnykov Serhii, Karnatskyi Dmytro, Bychkov Oleksii	Information-analytical system of electronic voting using the Fujioka-Okamoto-Ohta protocol	106

СОДЕРЖАНИЕ

Козачок П.А., Бражененко М.Г., Шевченко В.Л.	Облачно-ориентированный дизайн системы систем	15
Журко А. А., Шевченко В. Л.	Фильтр Калмана	16
Петракивский Д.А., Шевченко В.Л.	Кластеризация недвижимости на основе цены и особенностей расположения	17
Кузьменко Ольга, Поперешняк С.В.	Определение облака с помощью сворачиваемых нейронных сетей	18
Бычков А.С., Берестов Д.С., Шевченко В.В.	Модели реальной жизни на основе клеточных автоматов	19
Ильин М., Василенко В., Шевченко В.Л.	Проект PEERGAZE управления мобильным устройством с помощью взгляда	22
Марчук В.В., Шевченко В.Л.	Проект «Temp-sensor»	24
Бычков А.С., Шевченко В.Л., Беляченко О.В.	Спираль развития инженерии программного обеспечения и менеджмента проектов программного обеспечения	26
Ляшуга Николай, Ковтун О.И.	GitLab в командной разработке	30
Самойленко Олег, Супрун О.Н.	Bitbucket для командной разработки программного обеспечения	32
Мирошник Алексей, Шевченко В.Л.	Распределенный инструмент управления исходным кодом Mercurial	35
Баснакаев Салават, Берестов Д.С.	Сервис GitHub	37
Ситко Денис, Юрчук И.А.	Microsoft Project программное обеспечение для управления проектами	41
Дронь Дмитрий, Меркулова Е.В.	Worksection - сервис управления проектами для бизнеса	43
Цехмейструк Андрей, Доценко С.И.	JIRA - средство отслеживания задач и управления проектами	46
Панасовский Дмитрий, Ковалюк Т.В.	Notion - единый инструмент для ведения личных и корпоративных дел и задач	52

Шаламай Елизавета, Поляков С.А	Microsoft Teams - унифицированная платформа для общения и сотрудничества	55
Шапенко Евгений, Юрчук И.А.	Органайзер и почтовый клиент MS Outlook	59
Рыбалочка Михаил, Супрун О.Н.	Confluence - программное обеспечение командной работы	61
Шестаков Андрей, Поляков С.А.	Программа телекоммуникаций Skype	63
Балацкий Дмитрий, Меркулова Е.В.	Dropbox - программное обеспечение командной разработки	66
Дяченко Игорь	Evernote - инструмент для командной разработки программного обеспечения	69
Ткаченко М.В., Федоренко Р.М., Шевченко В.Л.	AcceleratedSAP (ASAP)	72
Берестов Д.С., Курченко О.А., Покотило А.І.	Особенности разработки программного обеспечения для обеспечения защиты информации от несанкционированного доступа	76
Берестов Д.С., Курченко О.А., Покотило А.І.	Построение комплексной системы защиты информации в автоматизированных системах	79
Бражиненко Максим, Ляшуга Николай, Самойленко Олег, Дронь Дмитрий, Цехмейструк Андрей, Шаламай Елизавета, Беляченко О.В.	Анализ соответствия учебных дисциплин требованиям производства программного обеспечения	82
Нестеренко А.В., Нетесин И.Е., Полищук В.Б.	Информационно-аналитическая система территориального управления качеством атмосферного воздуха	84
Артеменко Е.В. , Мидоян Э.Р.	Обработка данных компьютерным зрением в информационном пространстве с помощью линейной фильтрации	86
Карандась Д.Д., Ткаченко М.В., Федоренко Р.Н.	Использование нейронных сетей при оценке качества ПО	87
Холявкин Иван, Ковалюк Т.В.	Автоматизация определения перечня дисциплин специальности «Инженерия программного обеспечения» на основе мотивационного письма студента	89
Соколов К.О., Гудыма О.П.	Вопросы обеспечения кибербезопасности в всемирной информационной среде	91

Петривский В.Я., Шевченко В.Л., Бычков А.С.	Использование 5G технологий в Интернете Вещей и Интернете Всего	93
Ващук И. А., Поперешняк С.В.	Автоматизация учета и мониторинга рабочего времени на предприятии согласно действующего законодательства	94
Соколенко П.Ю., Поперешняк С.В.	Система обработки данных парковки автомобилей, как часть Smart City	97
Поперешняк С.В.	Некоторые аспекты тестирования псевдослучайной последовательности малой длины	101
Прудченко А.А., Старкова С.В., Герасименко К.В.	Алгоритм сопоставления изображений за ключевыми точками при масштабировании и вращении объектов	104
Мирошников С.А., Карнацкий Д.О., Бычков А.С.	Информационно-аналитическая система электронного голосования с использованием протокола двух агентств Фудзиоки-Окамото-Охта	106

MSTIoE 2019-5
**5-th East European Conference on Mathematical Foundations and Software
Technology of Internet of Everything**

5-та Східно-Європейська конференція
Математичні та програмні технології Internet of Everything

Conference presentations
Очні доповіді конференції
19.12.2019, Kyiv - Київ



В цьому році оргкомітет дещо змінив формат конференції і розпочав роботу дистанційної конференції з очної секції.

Головуючий – Віктор Шевченко, кафедри програмних систем і технологій

Гості секції:

Klaus Sickinger – Head of Central Eastern Europe msg Plant AG, Austria

Ольга Беяченко – комерційний директор компанії Софт-Рейтинг Консалт, Київ.

Валерій Поліщук – директор Українського наукового центру розвитку інформаційних технологій МОН України



Kozachok P.A. & Brazenenko M.G. master students
Shevchenko V.L. Dr.Sc., professor
Department of Software Systems and Technologies
Faculty of Information Technology
Taras Shevchenko National University of Kyiv, Kyiv,
Ukraine



Cloud based architecture design of system of systems

To remain competitive, digital-first companies must build, deploy, and integrate applications faster than ever. Currently Enterprise Application Integration (EAI) mediation and façade function commonly achieved by ESB solutions, which hard to maintain, evolve and assess operational quality.

The problem of rapid market entry in our time is the most urgent task to work out. In our time, in order for the company to have a competitive edge over its opponents it needs to release an updated version of the product as soon as possible. Today, there are many offers from different manufacturers that allow you to quickly release products. But, as it is real, most of them are supported by overhead costs (development for the target platform, support of this target platform on individual products, etc.). In order to face the best user experience the users must be in close proximity to the servers that will serve them. As you might know, the user chooses the most convenient service. That is why it can be seen as a competitive advantage.

Currently virtually anyone can connect to the Internet through multiple devices. We are seeing a rapid increase in the number of resources connected to the network, this fact will inevitably affect our lives.

Controlling the information field is better than the "big purse" in our time, where the world is changing more and more rapidly every year.

In order to meet the needs of the enterprise in data processing speed, end-user privacy, more and more companies are coming to a move to a distributed project architecture where each component is more secure than one large monolith and has a throughput capacity of up to 1 million requests per second.

Continuous monitoring of the functionality used and local scaling of the services required for the calculation is an integral part of the task for modern system architecture designers.

The concept of micro-service approach provides a revolutionary architecture that relies mainly on lightweight container technology for virtualisation. This design enables the development of distributed computing software systems capable of achieving high QoS performance

To achieve effective end-to-end communication, network bandwidth may not be sufficient, or the information requires some sort of processing, one of the worst cases is the lack of precise location or the constant change of location of the end devices that need interaction.

We use a capillary architecture for the delivery and loading of services. It is important to note that the feature of containers is that they are stored as a file in the container registry (artefacts). So, if we build some intermediate level of Fog, it will load the containers from the registry and store them closer to the end devices. Then we will be able to download the necessary services on the Raspberry PI based devices and direct the cloud traffic to them. So some calculations, filters, and other analysis of the dynamic data of the Internet can all be directed from the cloud closer to the consumer.

Conclusion:

A design of System of Systems software based on cloud resources, which is able to handle EAI mediation and façade function as a part of larger eco-system and satisfy integration needs of CAD\CAE systems was developed.

Proposed a simplified decision making model to evaluate QoS of an architecture design. Researched and classified common implementation pitfalls and most possible ways to tackle them. Evaluation of QoS metrics show that cloud based architecture has better qualities than existing industry-wide used alternative ESB.

Proposed an interpretation approach, based on Harrington function, for aggregated QoS results, identified major advantages and disadvantages for both approaches.

Zhurko O. O. Second year bachelor student
Shevchenko V.L. Dr.Sc., professor
Department of Software Systems and Technologies
Faculty of Information Technology
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine



Kalman filter

Everyone is interested in accuracy, especially when it comes to measuring something. Sensor indicators are not enough for someone and with the help of using various formulas and algorithms, they are trying to make results more accurate. In this case, the Kalman filter is more suitable than others are. This algorithm became widespread in [navigation](#), [control](#) of vehicles, computer vision, spacecraft as also [signal processing](#) and [econometrics](#).

The Kalman filter consists of two parts namely prediction and adjustment. Prediction is based on data in the previous time stage. Adjustment is a comparison of prediction our data and real values from the sensor. This combination makes the algorithm very fast and efficient, allowing implementation in weak systems and devices.

What if we expand incidence of the filter. No less important area of our gadgets, more correctly batteries of them. Sure batteries lose their ability over time. Thereby gadgets charge surges happen. It is annoying. To solve this problem the filter must be modified.

Firstly, it needs to determine coefficients. Coefficient responsible for device modes (Q) and error coefficient (R), which will constantly deteriorate. Matrices are deleted from the filter. Therefore, it leads to simplification of formulas and there are no coefficient F is a state transition, H is an observation, B is a control matrix and identity matrix I has been replaced by one. Formulas have the following form:

Prediction:

$$\bar{x}_k = x_{k-1}$$

$$P_k^- = P_{k-1} + Q$$

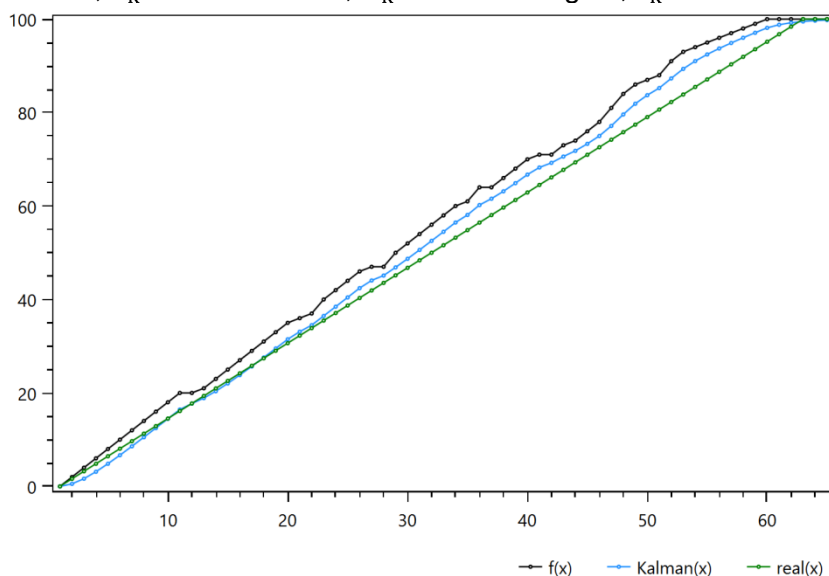
Adjustment:

$$K_k = P_k^- (P_k^- + R)^{-1}$$

$$x_k = \bar{x}_k + K_k (z_k - \bar{x}_k)$$

$$P_k = (1 - K_k) P_k^-$$

Where x_k is a state, P_k is a covariance, K_k is a Kalman gain, z_k is values from our sensor.



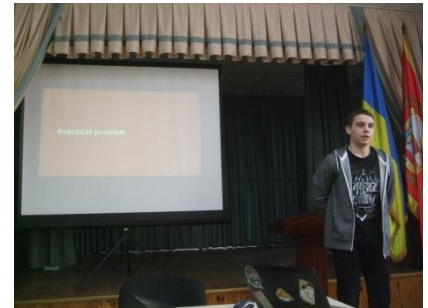
For example, if device is charging (Q is 1 and R is 5), a filter works better with a sensor than a single sensor. In addition, the total error on the indicator is 297.99 and on the Kalman filter is 143.81 (Manhattan assessment).

To sum up a new incidence of the Kalman filter is quite effective. It also makes sense to implement the algorithm at the system level because battery consumption lower than at the application. Devices will be more usable despite their useful life.

LITERATURE

1. <https://www.intechopen.com/books/introduction-and-implementations-of-the-kalman-filter/introduction-to-kalman-filter-and-its-applications>
2. <https://towardsdatascience.com/kalman-filter-an-algorithm-for-making-sense-from-the-insights-of-various-sensors-fused-together-ddf67597f35e>
3. <https://www.mathworks.com/videos/understanding-kalman-filters-part-1-why-use-kalman-filters--1485813028675.html>
4. <https://keldysh.ru/council/1/pankratov/pankratov-diss.pdf>

Petrakivsky D.O. 2 year bachelor student
Shevchenko V.L. Dr.Sc., professor
Department of Software Systems and Technologies
Faculty of Information Technology
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine



Real estate clustering, based on the price and placement features

Keywords: realty, data analysis, clustering, *kmeans*, *cmeans*, python, scikit-learn, postgres, postgresql, price zones, infrastructural features.

Practical problem. Nowadays real estate takes a significant place in our lives. Everyone wants to live at a comfortable flat or house, with efficient heating in the wintertime and cool conditioning in the summertime. Contemporary age brought us huge data massives in the field of realty as well. That's why many realty agencies and urban planners would appreciate a tool, capable of analysing the housing data. What's more, this tool will be even better, if it can segregate the target territory into zones representing a discrete distribution of a continuous variable, like a square meter price.

Development goal. Implementation of a self-succinct data analyzer (along with a data miner), capable of splitting of the target city territory into non-intersecting places with the average price per square meter within them. Data miner explores public APIs and parses webpages in order to collect info about the specified realty at the specified locality, and the analyzer calculates identity of each housing in order to clusterize all realty by the average level of wealth. Among features taken into account are downtown distance, water closeness, surrounding rapid transit stations, airports' and harbours' vicinity, geographical coordinates, etc.

Algorithms used. Clustering has two alternative implementations - usual *kmeans* clustering and *fuzzy cmeans* grouping. Spatial division was made via *Voronoi diagram*, which splits the plane into convex non-intersecting polygons.

Conclusion. As a result of the realty research we supply a simple application, capable of clustering of the target city into self-succinct price zones visualized via choropleth maps. Also it can be easily integrated into web apps and existing models. This tool can be useful for realty agencies, banking organizations, city planners and software companies which are willing to optimize their business processes in the area of real estate.

Kuzmenko Olha 2-nd course bachelor student,
Information technology faculty
Taras Shevchenko National University of Kyiv, Kyiv
Popereshnyak S.V. Ph.D., assistant professor
Department of Software Systems and Technologies
Faculty of Information Technology
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine



Cloud detection using convolution neural network

The importance of clouds

Looking at a satellite image of our planet, the first thing that strikes the eye is clouds. Almost 70% of the Earth is covered by clouds at any time. But clouds are not simply clouds. They come in a wide variety of shapes and forms.

Atmospheric scientists are interested in cloud formations not (only) because of their beauty but because small changes in the way clouds organize matter a lot for the climate. In particular, shallow clouds reflect a significant portion of the incoming solar radiation back into space and thereby help to keep the planet cool. There are 4 types of clouds: sugar, fish, flower and gravel.

Convolution Neural Network (CNN)

There are a lot of algorithms that people used for image classification before CNN became popular. People used to create features from images and then feed those features into some classification algorithm like SVM. Some algorithm also used the pixel level values of images as a feature vector too. But CNNs can be thought of automatic feature extractors from the image. While if I use a algorithm with pixel vector I lose a lot of spatial interaction between pixels, a CNN effectively uses adjacent pixel information to effectively down sample the image first by convolution and then uses a prediction layer at the end.

Convolutional Neural Network (CNN) take this name from mathematical linear operation between matrixes called convolution. CNN have multiple layers; including convolutional layer, non-linearity layer, pooling layer and fully-connected layer. The convolutional and fully-connected layers have parameters but pooling and non-linearity layers don't have parameters. The CNN has an excellent performance in machine learning problems. Specially the applications that deal with image data, computer vision, and in natural language processing (NLP) and the results achieved were very amazing.

Cloud detection using CNN

The images were downloaded from [NASA Worldview](#). Three regions, spanning 21 degrees longitude and 14 degrees latitude, were chosen. The true-color images were taken from two polar-orbiting satellites, TERRA and AQUA, each of which pass a specific region once a day.

The program identify regions in satellite images that contain certain cloud formations using convolution neural networks. Using this architecture, the program achieved high accuracy and low loss function. Program consists:

- GUI;
- Photo processing;
- CNN (3 convolution, 3 pooling, 3 fully connected layers);
- DataGenerator.

REFERENCES

1. <https://www.zooniverse.org/projects/raspstephan/sugar-flower-fish-or-gravel>
2. 'Understanding of a Convolutional neural network' – ICET2017, Antalya, Turkey

Shevchenko V.V., 1-st course bachelor students,
Bychkov O.S. Dr.Sc., professor, Head of department
Berestov D.S., PhD, assistant professor
Department of Software Systems and Technologies
Faculty of Information Technology
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine



Real life models with cellular automata

Introduction

In 1970th John Conway created cellular automaton Game of Life. He made this model to show, that even models with few variables can generate complex structures like computer or self-creating cell groups [1]. At the time, this game attracted a lot of attention from scientists around the world, and continues to attract even today. So I decided to create upgraded automaton, that would be able to model real life situations. When I ran into this game, I wanted to know if there were other available variations on this cellular automaton. I was surprised that I couldn't find a program to realize a game that would have a universal algorithm for all the variants of the input parameters. So I decided to write my own algorithm.

Classic "Game of Life" and its rules

The game takes place on an infinite cellular field. Each cell has 8 neighbors. If an empty cell has 3 living neighbors, the cell will have life in next step. If there are two or three living neighbors in a filled cell, the cell will have life in next step.

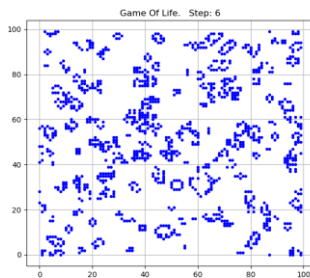


Fig. 1. Realization of classic Game of Life

Upgrades

In my work, I have made the following improvements to make the cellular machine more realistic:

- All situations algorithm
- Cell field connection in the torus
- Many populations models
- Neighbour array model
- Death by Age model
- Three dimension "Game of Life"

MatLab and Python realizations

When I created my universal algorithm, I used the MatLab programming language because it makes writing algorithms easier and faster. But later on, I also rewrote the algorithm to Python to extend the range of improvement tools.

Torus algorithm

The second change in the classic game was the connection of opposite corners and sides of the field between each other. Connected cellular field is equal to the torus [2]. The improvement required changes in the neighborhood function.

Victim – Predator model

One of the models that was found in the literature about cellular automata is the predator–victim model. In our model, in addition to previous changes, two new rules also apply. The first rule is that a predator eats a victim if they are in the same cell. The second rule: if there are not enough victims within a certain radius, the predator dies. The second rule is the predator

hunger rule. It balances out the populations in the model. Additionally, we also enter a graph of the number of each population on a time-dependent basis(Fig.3).

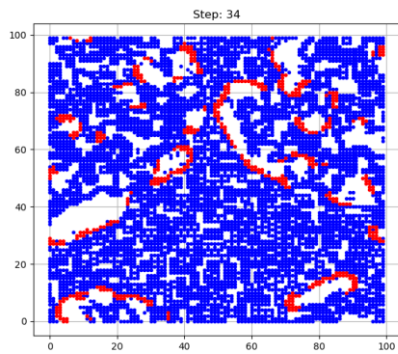


Fig. 2. Realization of Victim - Predator model.

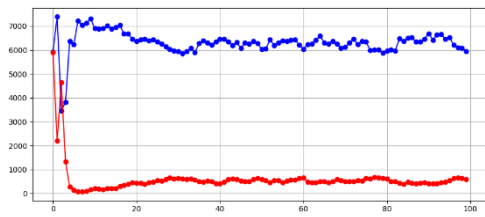


Fig. 3. The graph of the dependency of population size on time.

Neighbor array model

For better visibility of the model with one population, a model was created, which shows the number of neighbors in each cell in the graph(Fig.4). The brighter the color of a cell, the more neighbors a cell has. If the cell is white, it's alive.

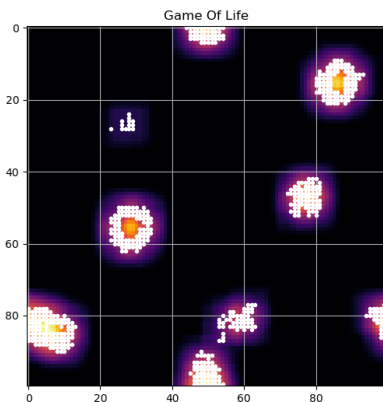


Fig. 4. Realization of neighbor array model

Death by Age model

In this model we introduce the next rule: a cell that does not stay empty more than a certain number of iterations dies. This addition of rules makes the model more dynamic. To plot a graph(Fig.5), we use the same function as in the model of the neighbor array.

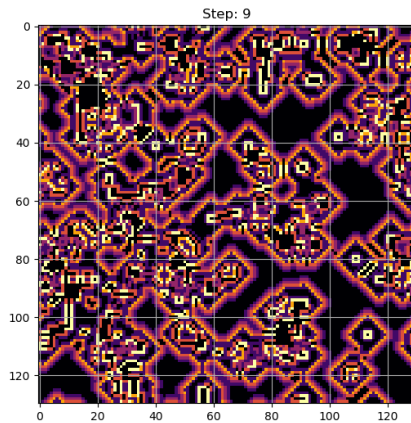


Fig. 5. Realization of Death by Age model

Three-dimension Game of Life

In the last model we implement a cellular automaton in a three-dimensional field. The changes in the code consist in more connections of opposite parts of the cube. The result of the connections is a hypertorus [2]. The algorithm is realized in MatLab(Fig.6), because this programming language is more suitable for working with arrays than Python.

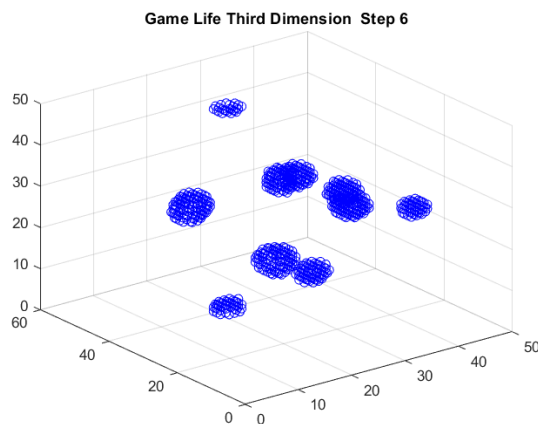


Fig. 6. Realization of Three-dimension Game of Life

Application

- These models can be used to simulate the spread of panic in a crowd.
- As a more general model, can be applied to modelling dissemination of conflicting information.
- Also for modeling population movements in nature.
- It is also possible to use in cryptography.
-

REFERENCES

1. Stephen Hawking, Leonard Mlodinow. The Grand Design. New York, NY, USA: Bantam Books, 2010, pp. 187–196.
2. Torus. <https://en.wikipedia.org/wiki/Torus>
3. Астафьев Г.Б., Короновский А.А., Храмов А.Е. Клеточные автоматы: Учебно-методическое пособие. Саратов: Изд-во ГосУНЦ «Колледж», 2003. 24с.
4. Andrew Adamatzky, Ramon Alonso-Sanz, Anna Lawniczak, Genaro Juarez Martinez, Kenichi Morita, Thomas Worsch. Theory and Applications of Cellular Automata. United Kingdom. Luniver Press, 2008, 636p.

Ільїн М. студент магістратури
 Василенко В. студент магістратури
 Шевченко В.Л. д.т.н., професор
 кафедра Програмних систем і технологій
 Київський національний університет імені Тараса
 Шевченка, м. Київ, Україна

Проект PEERGAZE керування мобільним пристроєм за допомогою погляду

Основні принципи роботи компоненту

Після отримання даних щодо відстеження напрямку погляду користувача, вони передаються до компоненту моделювання та керування пристроєм.

Вся система цього компоненту спроектована на принципах реактивного програмування, тому є дуже швидкодіючою та оптимальною як на етапі супроводу системи, так і на етапі розробки.

Для моделювання поведінки напрямку погляду користувача, відстеження точки фокусу погляду, роботи системи «жестів поглядом» та надання команд пристрою було обрано фреймворк Unity.

Хоча Unity вважається досить «важким» фреймворком для розробки та моделювання 3D контенту, його було обрано завдяки обсягу підтримуваних платформ, великому набору інструментів для моделювання та легкій інтеграції нативного коду.

Завдяки оптимізації нашого алгоритму відслідковування напрямку погляду, ми можемо з легкістю використовувати цей фреймворк одночасно з системою відслідковування напрямку погляду у реальному часі, що до цього часу не було реалізовано в жодному проекті.

Зв'язок системи з нативним компонентом

Для обміну даними необхідно було реалізувати інтерфейс, який дозволить передавати компонентіві актуальні дані про стан напрямку погляду та обличчя користувача у кадрі.

Для цього було розроблено інтерфейс на C, що дозволив нативному коду iOS викликати методи компонента моделювання та передавати актуальні дані.

Компонент приймає дані щодо напрямку погляду у вигляді двох нормалізованих векторів (x_1, y_1, z_1) та (x_2, y_2, z_2) для лівого та правого ока. Також система відстеження погляду передає координати положення обличчя в тривимірному просторі відносно камери пристрою у вигляді (x_3, y_3, z_3) .

Окрім цього, також між системами передається службова інформація про стан компонента, та команди запуску/відключення систем.

Принцип роботи відстеження точки фокусу погляду користувача.

У першу чергу, отримані дані від компоненту обробляються для побудови векторів напрямку у моделюванні фреймворку Unity завдяки вбудованій структурі **Vector3**.

Потім будується серединний вектор, для визначення вектору погляду з обох очей, бо дані про напрямок погляду можуть бути неточними. Графічно це показано на Рис. 1.

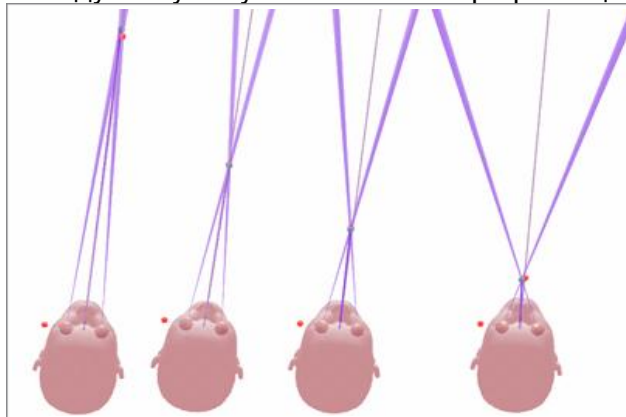


Рис. 1. Побудова серединного вектору напрямку погляду

Визначивши відстань обличчя користувача до екрану по координатам у обличчя у просторі, що передає компонент відслідковування напрямку погляду, система буде готова до калібрування.

Система автоматично визначає модель мобільного пристрою запитом до системи, та визначає фізичний розмір його екрану. Якщо користувач захоче калібрувати систему у ручному режимі, то йому буде запропоновано вибрати модель свого пристрою. Так як сьогодні на ринку запропоновано лише три різні версії iPad (за розмірами екранів у них), таких як iPad mini / iPad Air / iPad Pro із розмірами екранів 7,9 / 9,7 та 12.9 дюймів відповідно, то вибір буде запропоновано з них. Також, є можливість задати розміри екрану пристрою вручну.

Після цього для калібрування системи потрібно закріпити планшетний пристрій статично відносно голови користувача, та дивитись у центр мобільного пристрою кілька секунд. За цей час система запам'ятає набір напрямків погляду до центру пристрою та визначить середній з них, який буде опорним.

Для визначення точки фокусу погляду користувача на екрані вимірюється відхилення поточних координат нормалізованого середнього вектору погляду відносно опорного, будується промінь у 3D середовищі, та за умови перетинання його з умовним екраном пристрою, малюється точка фокусу на інтерфейсі

Принцип керування мобільним пристроєм та надання команд

Наступним кроком у розробці проекту було визначення методу розпізнання команд, та найоптимальнішого методу вводу команд поглядом.

Серед усіх підходів до можливого надання команд користувачем було розроблено новий підхід, що впровадив найпродуктивніший метод розпізнання простих рукописних знаків при рукописному вводі в умовах вводу напрямком погляду користувача. Розробка та впровадження такого методу була зумовлена також обмеженими обчислювальними ресурсами системи, бо використовувати ще одну нейронну модель для розпізнання намальованих символів у реальному часі нажалі неможливо.

Алгоритм цього методу є досить простим та ефективним, бо символи які можна малювати очима є досить примітивними.

Графічно алгоритм зображено на Рис. 2.

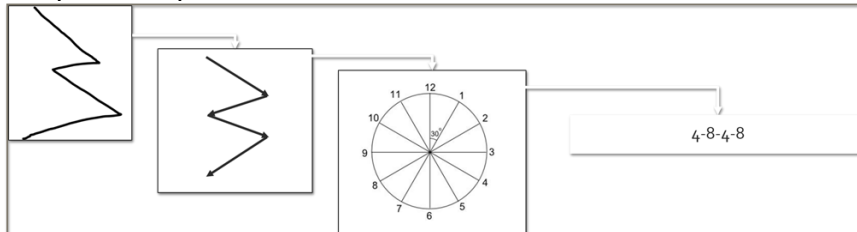


Рис. 2. Графічне зображення алгоритму розпізнання мальованих поглядом символів

Поетапно, алгоритм працює наступним чином:

- Спочатку крива, намальована поглядом розбивається на нормалізовані вектори;
- Координати початку цих векторів приводяться до початку координат;
- Уся координатна площа розбивається на 12 секторів з кутом відхилення у 30 градусів;
- Визначається якому з секторів належить кожен нормалізований вектор;
- Фігура записується у пам'ять як послідовність номерів секторів, та порівнюється з наборами заздалегідь запрограмованих команд;
- Якщо програма знаходить збіг, то виконується потрібна команда.

Для реалізації «Кліку» нами було запропоновано використовувати різноманітні Bluetooth гарнітури, що дозволить користувачам з обмеженими можливостями без проблем користуватись нашою системою через різноманітність вибору таких приладів

Марчук В.В. Студент 1 курсу
Шевченко В.Л., д.т.н., проф., професор кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса
Шевченка, м.Київ, Україна

Проект «Temp-sensor»



Температура – доволі важливий показник у роботі більшості техніки, ігнорування якого може призвести не тільки до виведення зі строю машини, але й до економічних проблем компанії чи втрати даних. Що ж такого може статися, якщо не слідкувати за температурою в технічному відділенні. Розглянемо звичайний сервер, його нормальна робота потребує температури в діапазоні від 0 до 16 градусів по Цельсію. Кімнатна температура 24 градуси є набагато вищою за норму. При високій температурі сервер може швидко вийти з ладу та навіть втратити дані. Візьмемо іншу ситуацію, чи звертали Ви увагу на роботу (кулерної) системи охолодження Вашого комп'ютера при низькій та високій температурах в кімнаті. Якщо в кімнаті висока температура то для охолодження центрального процесора потрібно більше часу. Звісно є ще дуже багато чинників які можуть впливати на охолодження, я навів доволі неточний приклад, так як ми говоримо за температуру навколишнього середовища.

До Вашої уваги АМ-В1 «Бутерброд», комп'ютер, що фіксує температуру та вологість повітря у крайніх точках Моніторної зони та знаючи залежності й фактори, що впливають на температуру, будує термокарту, з повною інформацією про небезпечні фактори в тому чи іншому секторі. У разі виходу температури за допустимі межі комп'ютер відправляє сповіщення на комп'ютер-моніторинг, вмикає червону індикацію та звукове супроводження. Але на цьому можливості Бутерброда не закінчуються, ми можемо запрограмувати його на автоматичне виконання протоколу, щодо усунення ситуації.

Створимо ситуацію. Осінь, Серверний відділ компанії, яка вирішила не витратити кошти на систему термо-контролю. Вирішила, що їй не потрібен АМ-В1 «Бутерброд». Настало літо, температура у відділі підвищилась. На серверах – головна база даних компанії. І тут неочікувано сервер вимкнувся. Доступ до бази даних втрачено, компанія не може переглянути список замовлень, але це ще невелика проблема, адже всі проекти також були записані на диск сервера. Звісно технічний спеціаліст запусте сервер знову, можливо він отримає інформацію, щодо причини аварійного вимкнення, де виявить, що температура компонентів вище допустимої. Але компанія вже втратила декількох клієнтів, які були дуже ображені на те, що під час замовлення отримали помилку. Технічний спеціаліст звернувся до менеджера та передав, що серверам потрібна система термо-контролю. Компанія закупила систему охолодження та АМ-В1 «Бутерброд». І знову температура за межами норми. Бутерброд виявив порушення умов, повідомила про ситуацію технічного спеціаліста та активувала систему охолодження. Температура стабілізувалась, сервери працюють добре, клієнти задоволені, працівник також, адже база даних працює добре.

Наразі продукт розвивається та в подальшому отримає дуже багато функцій, що нададуть технічним працівникам задоволення від використання саме АМ-В1 «Бутерброд».

MSTIoE 2019-5
5-th East European Conference on Mathematical Foundations and Software
Technology of Internet of Everything

5-та Східно-Європейська конференція
Математичні та програмні технології Internet of Everything

Remote conference reports
Дистанційні доповіді конференції
19-20.12.2019, Kyiv - Київ

Бичков О.С., д.т.н., доцент, завідувач кафедри
Шевченко В.Л., д.т.н., проф., професор кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна
Беляченко О.В.,
 комерційний директор компанії Софт-Рейтинг Консалт.

Спіраль розвитку інженерії програмного забезпечення та менеджменту проектів програмного забезпечення

Ми живемо в часи, що захоплюють динамікою розвитку технологій інженерії програмного забезпечення. Інколи ця динаміка здається космічно швидкою. А інколи ми зустрічаємо, ніби то, нездолані труднощі в рамках конкретного проекту. І коли класична інженерія програмного забезпечення не може надати гідних інструментів, на допомогу приходять суміжні галузі, зокрема менеджмент проектів програмного забезпечення. Спостерігається також зворотня картина. Менеджмент проектів програмного забезпечення все більше спирається на інструменти, що були народжені інженерією програмного забезпечення. Ці дві галузі утворюють спіраль розвитку в якій дві гілки розжениються одна одну.

Методологія командної розробки сьогодні грає одну з найважливіших ролей при створенні IT-проектів. Команди при створенні продуктів використовують все більше застосунків для полегшення сумісної роботи як в розробці, так й в менеджменті проектів.

Перша частині збірника тез присвячена саме програмним інструментам підтримки менеджменту проектів програмного забезпечення. Автори мають на меті описати в загальних рисах найбільш розповсюджені додатки, а саме їх функціональність, вартість та загальний досвід використання.

Короткий опис технологій. У даному збірнику розглянуті найбільш розповсюджені інструменти командної розробки, а саме:

- *GitLab* - зручний засіб контролю версіями програми, що також надає можливість зручного менеджменту задачами команди та подальшого запуску проекту. В основі лежить технологія репозиторіїв Git, що полегшує можливий перенос коду з інших платформ. Також цей сервіс дозволяє тримати як публічні, так і приватні репозиторії, використовуючи безкоштовну підписку. Також сервіс надає можливість "деплоїти" проекти у реальному часі, одразу ж після релізу.
- *Bitbucket* - веб-сервіс для хостингу проектів та їх спільної розробки, який заснований на системах контролю версій Mercurial та Git. За призначенням та основними запропонованими функціями є альтернативою GitHub, однак, на відміну від GitHub, який до січня 2019 року зберігав файли безкоштовних профілів лише у відкритому доступі, Bitbucket від самого початку дозволяв безкоштовно створювати приватні репозиторії з можливістю спільної роботи з файлами до 5-ти користувачів. Bitbucket інтегрований з іншими програмними продуктами Atlassian, такими як, JIRA, Confluence, Bamboo та інші.
- *Mercurial* - розподілений інструмент управління вихідним кодом. Він ефективно обробляє проекти будь-якого розміру і пропонує простий і інтуїтивно зрозумілий інтерфейс. Він простий у використанні і його важко зламати. Сервіс підтримує безліч робочих процесів, і ви можете легко розширити його функціональність за допомогою розширень. Реалізація Mercurial і структури даних спроектовані так, щоб бути швидкими.
- *GitHub* - сервіс онлайн-хостингу репозиторіїв, що володіє всіма функціями розподіленого контролю версій і функціональністю управління вихідним кодом - все, що підтримує Git і навіть більше. Зазвичай він використовується разом з Git і дає розробникам можливість зберігати їх код онлайн, а потім взаємодіяти з іншими розробниками в різних проектах.
- *Project* - це продукт від Microsoft, створений для управління проектами. Дане програмне забезпечення допомагає менеджеру проекту в розробці планів, розподілі ресурсів за завданнями, відстеженні прогресу і аналізі обсягів робіт. Project також дозволяє ефективно спілкуватися та співпрацювати з колегами.

Система управління проектами від Microsoft є простою у використанні, мобільною, оскільки має веб-додаток, тому для управління корпоративними проектами - це те що треба, оскільки ти завжди в курсі подій, а процес використання Project не викликає багато питань.

- *Worksection* - система дуже зрозуміла, клієнтоорієнтована, а якщо потрібна допомога техпідтримки, швидко вирішать питання. Дана система цікава з точки зору економічного та ефективного рішення. Функція проектного менеджера визначила витонченість взаємодії членів команди між собою і з відвідувачами. У випадку коли бізнес передбачає велику кількість клієнтів, і система не робить ніяких обмежень за цим профілем клієнтів, як і проектів або співробітників, може бути скільки завгодно.
- *Teams* - це сервіс від компанії Microsoft, головною метою якого є полегшення спілкування в командах, створений як альтернатива Slack і Atlassian's HipChat. У своїй найпростішій формі сервіс дозволяє користувачам створювати команди, (кожна з яких по суті є центром для групових чатів) які називаються каналами. Teams інтегровані в Інтернет-пакет Microsoft Office 365, що забезпечує зручну роботу з іншими службами Microsoft Office, такими як Word та Excel, а також з хмарними службами зберігання та обміну, такими як SharePoint, PowerPoint, OneNote, Planner. Окрім цього, Microsoft Teams дозволяє користувачам отримувати спільний доступ та редагування файлів, можливість перегляду календарів та легке перемикання між голосовими та відео дзвінками і текстовим чатом.
- *Microsoft Outlook* — клієнт для роботи з електронною поштою, який також є повноцінним органайзером. Електронна пошта, календар, контакти, завдання тощо доступні в одному місці для підвищення ефективності роботи. Для використання Outlook передбачений великий спектр різноманітних платформ, що робить його достатньо клієнтоорієнтованим. Outlook є мультиплатформним, тому його використання буде зручним в будь-якому місці, в будь-який час. Рекомендується для корпоративного використання.
- *Jira* - система відслідковування помилок, розроблена компанією Atlassian в 2002 році, призначена для організації взаємодії з користувачами та для управління проектами.
- *Notion* - сервіс, що об'єднує в собі величезну кількість інструментів. Нотатки та списки справ, документи та таблиці, канбан-дошки і бази знань - тут є все, з чим нам доводиться мати справу щодня. Сервіс являє все це у вигляді єдиного робочого простору. Подібно деталей LEGO, в нього легко додати потрібні елементи і зібрати свій ідеальний інструмент продуктивності для зберігання ідей, планування і спільної роботи з колегами. Notion дозволяє вести всі процеси в чистому, універсальному середовищі і тримати їх під контролем.
- *Confluence* - це зручний і просунутий wiki движок від компанії Atlassian. Він дозволяє організувати внутрішній інтернет портал і дати доступ до нього всім користувачам - для редагування або для читання.
- *Skype* – програма телекомунікацій, яка спеціалізується на наданні відеочату та голосових дзвінків між комп'ютерами, планшетами, мобільними пристроями. Користувачі можуть передавати текст, відео, аудіо та зображення. Skype дозволяє здійснювати відеоконференції.
- *Dropbox* - файловий хостинг компанії Dropbox Inc., що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт. Dropbox дозволяє користувачеві розміщувати файли на віддалених серверах за допомогою клієнта або з використанням веб-інтерфейсу через браузер.
- *Evernote* — онлайн-сервіс призначений для збереження, синхронізації та пошуку приміток, в тому числі текстових записів, веб-сторінок, списку задач, фотографій, малюнків, електронної пошти та ін. Evernote виконує функцію ефективного записки, який дозволяє дуже швидко зберігати інформацію (завдяки веб-технологіям) і фільтрувати (система тегів).
- *AcceleratedSAP (ASAP)* — це інструмент для швидкого впровадження проектів в системі SAP. Методологія ASAP – це структурний підхід до впровадження, який значно прискорює просування проекту і забезпечує ефективне навчання

користувачів, вичерпну документацію і ясно складені мережеві графіки на всіх стадіях проекту. В основу методології ASAP закладено жорстку каскадну модель – кожна фаза включає формальну процедуру запуску та формальну фазу завершення, що включає набір контрольних заходів.

Порівняльний аналіз програмних інструментів командної розробки програмного забезпечення наведений в табл.1.

Таблиця 1.

	GitLab	Bitbucket	Mercurial	GitHub	MS Project	Worksection	JIRA	Notion	MS Teams	MS Outlook	Confluence	Skype	Dropbox	Evernote
Кроссплатформеність	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Мобільний додаток	✓	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
WEB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PC	✓	✗	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Безкоштовна версія	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Ведення заміток	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
Корпоративна Вікі	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗
Синхронізація	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Інтеграція з іншими інструментами	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Складність використання	3/5	3/5	3/5	3/5	3/5	3/5	4/5	2/5	3/5	1/5	3/5	2/5	2/5	3/5
Управління проектами	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
Календар	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
Оповіщення	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Управління завданнями	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓
Управління знаннями	✗	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗
Моделювання бізнес-процесів	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Управління версіями	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓
Управління резервним копіюванням	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗

Автозбірки (інтеграція ПЗ)	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Тестування	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Розгортання	✓	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Набір персоналу	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Управління кар'єрою	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Звіти	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗

Ляшуга Микола, студент бакалаврату,
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна
співробітник компанії **Walkout Technologies GmbH**;
Ковтун О.І., к.ф.-м.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

GitLab в командній розробці

На сьогоднішній день, майже жодний з проектів не був би можливим без командної розробки. У команді більшість проблем можна вирішити набагато швидше, ніж самому. Але такий підхід створює величезні незручності при сумісному редагуванні спільного коду. Тому одним із основних рішень цієї проблеми є система керування версіями Git [1, 2].

GitLab - сервіс, що забезпечує віддалений доступ до сховищ Git. Окрім можливості розміщення вашого коду, сервіс надає додаткові функції, призначені для управління життєвим циклом розробки програмного забезпечення. Ці додаткові функції включають управління спільним кодом між різними людьми, відстеження помилок, вікі-простір та інші інструменти для командної розробки.

Функціональність

Робота з репозиторієм

Як було зазначено вище, GitLab надає зручний доступ та менеджмент програмного коду проекту. На головній сторінці проекту можна побачити два основних блока: меню доступу до відділів проекту (зліва) та загальну інформацію про код проекту (справа).

В правій частині можна побачити основну інформацію про проект. Задля взаємодії з проектом, кожен етап збереження коду представляється у вигляді **комітів**. Коміт – це одиниця збереження даних в Git, яка містить в собі конкретні зміни в коді та назву, яка описує загально основні нововведення.

Також для введення розробки коду, кожен з учасників має змогу створити **гілку коду**. Гілка представляється у вигляді окремої робочої середовища, при створенні якої містить останні зміни. Використовуючи гілки, девелопери мають змогу продовжувати вносити зміни в код, не змінюючи при цьому код інших розробників.

Для внесення змін та додавання нових змін, проект спершу треба відтворити на робочу станцію виконуючи команду **Clone**. Основною командою є: `git clone <лінк на клонування проекту з відділу Clone у верхньому правому кутку>`.

Отримавши останню версію проекту можна вносити зміни в деякі частини коду та виконати певні команди, наведені нижче, для збереження їх у іншу гілку.

Для прикладу ми проведемо зміни у файлі README, де зберігається пояснювальна частина функціоналу проекту.

Спочатку треба виконати команди по встановленню нової гілки : `git checkout -b <назва гілки>`.

Надалі вносяться зміни, та виконується реєстрація змін у репозиторії за допомогою таких команд:

- `git add .`
- `git commit -m "<Назва коміту>"`
- `git push`

Окрім цього, є можливість розмежування можливостей доступу до проекту між девелоперами. Використовуючи такий підхід, можна запобігти неправильне втручання в робочий код випадково.

Розподілення задач

Задля правильного ведення командної розробки, кожному з розробників має бути виділені одна чи більше **задач ("Issue")**. Задача включає в себе назву, детальний опис, відповідальних людей, дату та інколи складність у вигляді певного числа.

Використовуючи задачі, можливість чітко відслідковувати прогрес вирішення тієї чи іншої проблеми.

Збірка проекту

Використовуючи методологію деплоінгу проектів CI/CD (“Continuous integration and continuous delivery”, можна пришвидшити релізу коду в продакшн оминаючи мануальний запуск проекту. Такий підхід автоматизує подальшу роботу з кодом, та полегшує контроль роботи DevOps-ом.

Вартість. GitLab має на меті надавати свій сервіс якомога більшій кількості людей. І таким чином, комерціалізація проводиться у вигляді річної підписки на сервіс. В GitLab є такі види підписок:

- *Безкоштовна* (0) – надає можливість створювати необмежену кількість проектів та репозиторіїв, та надає базовий функціонал CI/CD.
- *Bronze* (4 \$ на місяць) – всі функції *Безкоштовної*, а також службову підтримку, можливість багато-юзерного схвалення зведення гілок.
- *Silver* (19 \$ на місяць) – всі функції *Bronze*, а також додаткові інструменти Scrum (Canary board тощо), загальний вигляд усіх задач між групами та збільшена кількість деплоїв в CI/CD до 10,000.
- *Gold* (99 \$ на місяць) – всі функції *Silver*, а також можливість пошуку vulnerabilities в проекті, roadmap та збільшена кількість деплоїв в CI/CD до 50,000.

Тим не менш GitLab дає змогу розкривати системи репозитаріїв на власному сервері.

Час потрібний на опанування. Мінімальний поріг входження в роботу з системою не є надто високим. Хоча для початку можуть знадобитися базові навички володіння Git. В загальному, сервіс надає зручний і зрозумілий інтерфейс взаємодії. Тим паче, більшість з кроків по налаштуванню проектів комбіновані разом з підказками, які відповідають на більшість питань нового юзера системи. Але при виникненні будь-яких питань є можливість звернутися до загальної документації сервісу по такому гіперпосиланню: <https://docs.gitlab.com>.

Платформи та синхронізація. Для взаємодії та оновлення коду використовують базова система Git. І таким чином, для роботи з GitLab буде необхідно встановити основний додаток GitLab Runner, який підтримується на багатьох PC платформах: Windows, Mac, Linux, FreeBSD. Цей додаток надає змогу підключитися до репозиторіїв та проводити подальшу взаємодію з ними.

Інтеграція з іншими інструментами. GitLab позиціонує себе як один із один із популярніших сервісів для командної розробки, таким чином він надає можливість використовувати функціонал інших сервісів, а саме:

- автентифікацію за допомогою BitBucket;
- мануальне налаштування серверу CI/CD на основі платформи Jenkins;
- додавання запитів на зведення коду у вигляді Trello карт.

Висновок

GitLab – один із найбільших лідерів на ринку надання сервісу ведення репозиторіїв, разом з GitHub, BitBucket тощо. Система має достатньо плюсів, а саме повний доступ до Git функціоналу, ведення та моніторинг задач, налагодження процесі CI/CD. Враховуючи основні характеристики і функціонал, можна з легкістю стверджувати, що система об'єднує в собі майже всі сторони командної розробки і може надавати сервіс навіть у безкоштовному режимі.

ЛІТЕРАТУРА

1. <https://docs.gitlab.com/ee/user/index.html> - GitLab User Docs
2. https://www.tutorialspoint.com/gitlab/gitlab_introduction.htm - GitLab Introduction

Самойленко Олег, студент бакалаврату,
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна
 співробітник компанії **GlobalLogic**;
Супрун О.М., к.ф.-м.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Bitbucket для командної розробки програмного забезпечення

Bitbucket - веб-сховище управління версіями, що належить компанії Atlassian. Використовується для зберігання версій вихідного коду та проектів розробки, які використовують Mercurial або Git.

Функціональність

Про гілки та запити на злиття

Працюючи в команді з декількома користувачами Bitbucket, ви хочете працювати над власним набором коду окремо від основної кодової бази. Гілки дозволяють робити саме це. Гілка являє собою незалежну лінію розвитку вашого сховища. Подумайте про це як про новий робочий каталог, зону постановки та історію проекту. Після створення гілки, працюйте на ній, модифікуйте цієї гілки, витягуйте оновлення з Bitbucket, щоб оновити версію проекту, а потім відправте всю свою роботу на Bitbucket.

Після того, як ви зміните код на гілці в Bitbucket, ви можете створити запит на злиття, де відбувається перевірка вашого коду. Ваші товариші по команді коментуватимуть ваш код за допомогою коментарів та запитань та врешті-решт схвалити запит на злиття. Коли у вас є достатня кількість схвалень, підтвердіть запит на злиття, щоб об'єднати свою гілку та основний код.

Якісне програмне забезпечення завдяки перевірці коду

Використовуйте запити pull, щоб ефективніше затверджувати зміни коду. Створіть контрольний список відповідальних людей для злиттів, і обговорюйте питання прямо в вихідному коді за допомогою вбудованих коментарів.

Часте розгортання завдяки вбудованим можливостям безперервної поставки

Можливість розгортання в Bitbucket Pipelines дозволяє виконувати збірку, тестування і розгортання коду з використанням вбудованого конвеєра безперервної інтеграції та безперервного постачання (CI / CD). Використовуйте всі переваги конфігурації коду і швидкого зворотного зв'язку.

Безпека робочого процесу

Завдяки прозорим списками IP-адрес і обов'язковій двофакторній аутентифікації Ви будете впевнені в захищеності свого коду в хмарі. Щоб створювати якісний код, надавайте доступ тільки певним користувачам і контролюйте їх дії за допомогою прав доступу до гілок і перевірок злиттів.

Bitbucket Server. Повністю контролюйте свій вихідний код за допомогою можливостей Bitbucket для розміщення на власному хостингу. По мірі зростання команди і розвитку робочого процесу можна виконати перехід від розгортання на одному сервері до цілого кластеру на базі Bitbucket Data Center.

Вартість. Додаток розповсюджується методом платної підписки. Спробувати можна безкоштовно, але є обмеження на об'єм завантажуваних даних. При звичайному користуванні швидко досягається ліміт безкоштовної версії. Варіанти підписок:

1. Free – \$0 – 1000 ліміту на кількість блоків
2. Personal - \$4/місяць – ідеально для персонального користування, адже відкриває всі можливості додатку без лімітів окрім користування в команді.
3. Team - \$8/місяць (кожен з команди) – підходить для проектних команд невеликого розміру.
4. Enterprise - \$20/місяць (кожен з команди) – для великих організацій.

- Безкоштовний необмежений обсяг закритих репозитаріїв
- Безкоштовно для невеликих команд (до 5 учасників)
- При подальшому розширенні можна оформити план Standard (3\$ за користувача в місяць) або план Premium (6\$ за користувача в місяць)

Час потрібний на опанування. Сервіс є інтуїтивно зрозумілим для користувачів, які мали досвід роботи з git та іншими системами контролю версій. Проте, для новачків Bitbucket надає велику кількість туторіалів щодо git, mercurial, а також документацію про взаємодію з гілками, pull-запитами, Bitbucket Cloud тощо.

Платформи та синхронізація. З сервісом можна працювати через звичайний браузер, тому він доступний для всіх платформ.

Інтеграція з іншими інструментами

Інтеграція з JIRA

- Автоматичні посилання коду на Джіру
 - Автоматичне зв'язування проблеми та інформування усієї команди
 - Створення гілки за допомогою Jira
 - Редагування проблем Jira всередині Bitbucket
 - Отримайте контекст своєї роботи та оновіть свою команду без переключення програм. Переглядайте, редагуйте, коментуйте тікети на Jira в інтерфейсі Bitbucket Cloud
 - Дивіться інформацію про розвиток в Джірі
 - Оцініть статус розвитку проблеми Джіра з першого погляду.
- Переглядайте та створюйте гілки та витягуйте запити та переглядайте review, не залишаючи Джіру

Інтеграція з Bamboo Server

Побудувати, протестувати, розгорнути. Bamboo Server - це вибір професійних команд для постійної інтеграції, розгортання та доставки. Почати роботу можна безкоштовно.

- Зв'яжіть автоматизовані збірки, тести та випуски разом в одному робочому процесі
- Побудова - зосередьтеся на кодуванні та використовуйте Bamboo як на свій CI та сервер побудови.
- Створіть плани багатоетапної збірки, встановіть тригери, щоб почати нарощування на комітетах, і призначте агентів для своїх критичних збірок і розгортань.
- Тестування є ключовою частиною постійної інтеграції. Запустіть автоматичні тести на бамбук, щоб ретельно регресувати свої продукти з кожною зміною. Паралельні автоматизовані тести розкривають потужність Agile Development та роблять помилки помилок легшими та швидшими.
- Розгортання - Bamboo пропонує першокласну безперервну підтримку хмарного розгортання.

Інтеграція з Trello

- Дошки Trello в Bitbucket
- Просте відстеження проекту - дошки, списки та картки Trello дозволяють вам швидко, гнучко організовувати, визначати пріоритети та співпрацювати над своїми проектами.
- Все в одному місці - ваші відставання та код у всіх місцях, тому вам ніколи не потрібно залишати Bitbucket, щоб знайти наступну роботу. Плануйте, відстежуйте, співпрацюйте та розгортайте, все з Bitbucket.
- Пов'язати код із завданнями - спростіть робочий процес, створивши нові гілки з карток і швидко переходьте до кодування. Слідкуйте за тим, щоб проекти були організовані, і всі були в циклі, пов'язуючи гілки, виконуючи чи перетягування запитів на картки та забезпечте завершення всієї вашої роботи.
- Отримайте статус з першого погляду - швидко зрозумійте стан роботи команди в простому, засвоєваному вигляді: з кольоровими позначками карток, статусів, запитів та збірок.

Рекомендована галузь використання. Bitbucket - це не просто інструмент управління кодом в Git. Bitbucket надає єдиний центр для планування проектів, сумісної роботи над кодом, тестування та розробки **для маленьких та великих команд**.

Команди Bitbucket Cloud дозволяють керувати кількома проектами та сховищами та співпрацювати з користувачами під час створення дивовижного програмного забезпечення.

З командою ви можете:

- Створювати та надавати доступ до сховищ вашої команди.
- Створювати проекти, щоб організувати та зосередити роботу своєї команди.
- Встановлювати конкретні дозволи для різних ролей чи команд.
- Керувати витратами вашої команди за допомогою одного плану користувача.
- Інтегрувати проекти своєї команди з такими послугами, як Jira Software та Slack.

Як працюють команди:

- У вашій команді є різні компоненти: користувачі, групи та сховища.
- Ваша команда володіє вашими сховищами.
- Усі команди мають групи користувачів. Групи задають дозволи за замовчуванням

для сховищ вашої команди.

- Додайте або запрошуйте користувачів до груп команди, надаючи їм права доступу до сховищ.

- Групуйте користувачів та налаштовуйте доступ

• У кожній команді за замовчуванням є дві групи користувачів, група користувачів-адміністраторів та група користувачів-розробників.

- Групи команди були додані до сховища зі своїми дозволами за замовчуванням, але ви можете будь-коли змінити доступ до групи.

- Якщо член команди має дозвіл на створення командних сховищ, цей користувач стає адміністратором для кожного створеного ними сховища.

Висновок. Bitbucket, як сервіс для контролю версій вихідного коду, схожий на GitHub, який в першу чергу використовує Git. Компанія Bitbucket традиційно продає свої послуги професійним розробникам із приватним програмним кодом, особливо з моменту придбання Atlassian у 2010 році. Bitbucket стабільно зростає: у лютому 2017 року Bitbucket оголосив, що досяг 6 мільйонів розробників та 1 мільйон команд на своїй платформі, а у квітні 2019 Atlassian оголосив, що Bitbucket досяг 10 мільйонів зареєстрованих користувачів та понад 28 мільйонів сховищ.

У Bitbucket є три моделі розгортання: Cloud, Bitbucket Server і Data Center, що дає йому перевагу у порівнянні з іншими сервісами для контролю версій.

Текст написаний на підставі особистого досвіду автора, а також з використанням таких джерел [1, 2, 3,4].

ЛІТЕРАТУРА

1. <https://bitbucket.org/product/guides> - Bitbucket
2. <https://en.wikipedia.org/wiki/Bitbucket> - Вікі Bitbucket
3. <https://www.atlassian.com/git/tutorials> - посібник по atlassian
4. <https://bitbucket.org/blog/how-i-use-bitbucket-in-my-regular-routine> - Як я використовую Bitbucket в моєму звичайному житті

Мірошник Олексій, студент бакалаврату;
Шевченко В.Л., д.т.н., проф., професор кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Розподілений інструмент управління вихідним кодом Mercurial

Mercurial SCM - це безкоштовний, розподілений інструмент управління вихідним кодом [1,2,3]. Він ефективно обробляє проекти будь-якого розміру і пропонує простий і інтуїтивно зрозумілий інтерфейс. Він простий у використанні і його важко зламати, що робить його ідеальним для тих, хто працює з версійними файлами.

Традиційні системи контролю версій, такі як Subversion, являють собою типові клієнт-серверні архітектури з центральним сервером для зберігання ревізій проекту. Навпаки, Mercurial дійсно поширюється, надаючи кожному розробнику локальну копію всієї історії розробки. Таким чином, він працює незалежно від доступу до мережі або центрального сервера. Фіксація, розгалуження і злиття швидко і дешево.

Функціональність. Mercurial у першу чергу є консольною програмою. Репозиторії Mercurial управляються при помощи утилити командної строки hg, но есть и GUI-интерфейсы. Система Mercurial написана на Python, хотя чувствительные к производительности части (например, своя реализация diff) выполнены в качестве модулей-расширений на C. Поддерживается обмен данными между репозиториями через HTTP/HTTPS, SSH и вручную при помощи упакованных наборов изменений. Утилита hg обладает компактным интерфейсом, и Mercurial считается более простой в освоении системой, чем, например, git.

Розподілена (децентралізована) система керування версіями

Порівняно новий напрям систем керування версіями. Від традиційних, централізованих систем керування версіями децентралізовані відрізняються:

- Кожний розробник працює з власним репозиторієм.
- Модель роботи схожа на «базарний» стиль в тому, що кожен може створити власні гілки.
- Репозиторії можуть клонуватись будь ким, і часто клонуються декілька раз.
- Може існувати декілька «центральных» репозиторіїв.
- Обмеження прав доступу не використовуються. Натомість, зміни додаються на основі мереж довіри, наприклад, на основі попереднього досвіду співпраці або якості змін.
- Важелі керування належать керівникам проектів, котрі вирішують які гілки об'єднувати.
- Більшість операцій не потребує мережевого з'єднання.
- Існує перелік операцій для синхронізації репозиторіїв — передачі або отримання змін з інших репозиторіїв.

Консольна програма реалізована таким чином, що назва будь-якої команди можна скорочувати до тих пір, поки її ім'я залишається однозначним. Плюс деякі команди мають псевдоніми. Наприклад, замість hg commit можна написати hg commi, hg comm або hg com, але якщо написати hg c, то Mercurial відмовиться виконувати цю команду, повідомивши, що «команда 'c' неоднозначна» і привівши список команд, які потрапляють під це скорочення. Використовувати hg co як скорочення для hg commit не можна, так як це псевдонім для команди hg update, проте є скорочення hg ci.

Mercurial є розподіленою (децентралізованою) системою контролю версій. Це означає, що **робочий процес**, як правило, виглядає наступним чином:

1. На особистому комп'ютері створюється новий репозиторій (шляхом клонування існуючого сховища, створення нового і т. п.);
2. У робочому каталозі даного сховища змінюються / додаються / видаляються файли;
3. Виконується фіксація (commit) змін в даний репозиторій (тобто в локальний репозиторій на особистому комп'ютері);
4. Кроки 2 і 3 повторюються стільки разів, скільки необхідно;

5. При необхідності проводиться синхронізація змін з іншими репозиторіями: забираються (pull) чужі набори змін та / або віддаються (push) власні.

Тобто вся повсякденна робота відбувається в локальному репозиторії, а коли виникає необхідність, проводиться відправка результатів своєї роботи в один або кілька інших репозиторіїв. Кількість кроків при роботі з віддаленими репозиторіями можна скоротити, якщо налаштувати Mercurial на автоматичну відправку змін в інші репозиторії при виконанні фіксації.

Вартість. Mercurial - безкоштовний інструмент управління вихідним кодом.

Час потрібний на опанування. Мінімальний поріг входження в роботу з системою не є надто високим. Хоча для початку можуть знадобитися базові навички володіння Git. Модель децентралізованої розробки Mercurial может сбить с толку новых пользователей: але для новачків Mercurial надає велику кількість туторіалів та мануалів щодо установки та використання системою.

Платформи та синхронізація. Mercurial дуже простий та добре інтегрований з оболонкою Windows. Якщо ви розробник Windows, який шукає базовий відключений контроль редагування, перейдіть на Hg.

Інтеграція з іншими інструментами. Mercurial - це DVCS, який передає код між локальною системою та хмарою Bitbucket.

Надсилання змін електронною поштою може бути найпростішим способом зв'язатися з людьми, коли ви ще не входите до штатної команди розробників, але це створює додаткове навантаження: Ви повинні згрупувати зміни, надіслати пошту та імпортувати пакети вручну. На щастя, є простіший спосіб, який працює досить добре: спільне сховище push.

До цього часу всі зміни ми перенесли або електронною поштою, або через pull. Тепер ми використовуємо інший варіант: pushing. Як впливає з назви, це просто навпаки від pull: ви переміщуєте зміни до іншого сховища.

Але, щоб скористатися цим, нам спочатку потрібно щось, на що ми можемо підштовхнути.

За замовчуванням сервіс hg не дозволяє пушити (push), оскільки це було б головним отвором у безпеці. Ви можете дозволити пушити (push) на сервері, але це не вирішення, коли ви живете в різних часових поясах, тому ми підемо з іншим підходом тут: Використання спільного сховища або на існуючому спільному сервері, або на такій службі, як BitBucket. Це має трохи більшу стартову вартість і потребує трохи довшого пояснення, але це варто того, щоб витратити зусилля.

Висновок. Mercurial ефективно обробляє проекти будь-якого розміру і виду. Кожен клон містить всю історію проекту, тому більшість дій локально, швидко і зручно. Сервіс підтримує безліч робочих процесів, і ви можете легко розширити його функціональність за допомогою розширень.

Реалізація Mercurial і структури даних спроектовані так, щоб бути швидкими. Користувач може генерувати відмінності між ревізії або повернутися назад в часі протягом декількох секунд. Тому Mercurial ідеально підходить для великих проектів, таких як OpenJDK або NetBeans

Велика частина Mercurial написана на Python, а невелика частина - на які переносяться С з міркувань продуктивності. В результаті бінарні випуски доступні на всіх основних платформах.

ЛІТЕРАТУРА

1. https://courses.cs.ut.ee/LTAT.05.009/2018_spring/uploads/Main/Lecture8.pdf - презентація про Mercurial
2. <https://www.mercurial-scm.org/> - офіційний сайт Mercurial
3. <https://en.wikipedia.org/wiki/Mercurial> - стаття Вікіпедії про Mercurial

Баснакаєв Салават, студент бакалаврату;
Берестов Д.С., к.т.н., асистент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Сервіс GitHub

GitHub - сервіс онлайн-хостингу репозиторіїв, що володіє всіма функціями розподіленого контролю версій і функціональністю управління вихідним кодом - все, що підтримує Git і навіть більше [10, 11, 12]. Зазвичай він використовується разом з Git і дає розробникам можливість зберігати їх код онлайн, а потім взаємодіяти з іншими розробниками в різних проектах. Перед тим як говорити про GitHub, розберемо системи контролю версій.

Системи контролю версій (СКВ, VCS, Version Control Systems) дозволяють розробникам зберігати всі зміни, внесені в код. Тому в разі, описаному вище, вони можуть просто відкотити код до робочого стану замість того, щоб витратити години на пошуки маленької помилки або помилок, які ламають весь код.

СКВ також дають можливість декільком розробникам працювати над одним проектом і зберігати внесені зміни, щоб переконатися, що все можуть стежити за тим, над чим вони працюють. Існує три типи СКВ: локальна, централізована і розподілена.

Локальні системи контролю версій (ЛСКВ). Багато людей в якості методу контролю версій застосовують копіювання файлів в окрему директорію, можливо навіть в директорію з відміткою за часом для більшого контролю. Даний підхід все ще дуже популярний і поширений. Зміни зберігаються у вигляді наборів патчів, де кожен патч датується і отримує позначку часу. Таким чином, якщо код перестає працювати, набори патчів можна поєднати, щоб отримати початковий стан файлу.

Централізовані системи контролю версій (ЦСКВ). ЦСКВ були створені для вирішення проблеми взаємодії з іншими розробниками. Такі системи мають єдиний сервер, який містить всі версії файлів, і деяка кількість клієнтів, які отримують файли з цього централізованого сховища і там же їх зберігають. Проте, такий підхід має істотний недолік - вихід сервера з ладу обернеться втратою всіх даних. а ще в таких системах може бути ускладнена одночасна робота декількох розробників над одним файлом.

Розподілені системи контролю версій (РСКВ). Недолік ЦСКВ був виправлений в РСКВ, клієнти яких не просто скачують знімок всіх файлів (стан файлів на певний момент часу), а повністю копіюють репозиторій. Це означає, що у кожного клієнта є копія всього вихідного коду і внесених змін. В цьому випадку, якщо один з серверів вийде з ладу, будь-який клієнтський репозиторій може бути скопійований на інший сервер для продовження роботи. Ще однією перевагою РСКВ є те, що вони можуть одночасно взаємодіяти з декількома віддаленими репозиторіями, що означає, що ви можете паралельно працювати над декількома проектами.

Проаналізуємо **Git** і його відмінності від **GitHub**.

Git - розподілена система контролю версій, яка дає можливість розробникам відстежувати зміни в файлах і працювати спільно з іншими розробниками. Вона була розроблена в 2005 році Лінус Торвальдс, творцем Linux, для того, щоб інші розробники могли вносити свій вклад в ядро Linux. Git відомий своєю швидкістю, простим дизайном, підтримкою нелінійної розробки, повної децентралізацією і можливістю ефективно працювати з великими проектами. Git варто позиціонувати окремо від інших СКВ через підхід до роботи з даними. Більшість інших систем зберігають інформацію у вигляді списку змін в файлах. Замість цього, підхід Git до зберігання даних більше схожий на набір знімків мініатюрної файлової системи. Кожен раз, коли ви зберігаєте стан свого проекту в Git, система запам'ятовує, як виглядає кожен файл в цей момент, і зберігає посилання на цей знімок.

Переваги Git

- Безкоштовний і open-source. Це означає, що його можна безкоштовно завантажити і вносити будь-які зміни у вихідний код;
- Невеликий і швидкий. Він виконує всі операції локально, що збільшує його швидкість. Крім того, Git локально зберігає весь репозиторій в невеликий файл без втрати якості даних;
- Резервне копіювання. Git ефективний в зберіганні бекапов, тому відомо мало випадків, коли хтось втрачав дані при використанні Git;
- Просте розгалуження. В інших ВКВ створення веток- втомлива і трудомістка задача, так як весь код копіюється в нову гілку. У Git управління гілками реалізовано набагато простіше і ефективніше.

GitHub - сервіс онлайн-хостингу репозиторіїв, що володіє всіма функціями розподіленого контролю версій і функціональністю управління вихідним кодом - все, що підтримує Git і навіть більше. Зазвичай він використовується разом з Git і дає розробникам можливість зберігати їх код онлайн, а потім взаємодіяти з іншими розробниками в різних проектах

Також GitHub може похвалитися контролем доступу, багтрекінгом, управлінням завданнями і вики для кожного проекту. **Мета GitHub** - сприяти взаємодії розробників.

До проекту, завантаженому на GitHub, можна отримати доступ за допомогою інтерфейсу командного рядка Git і Git-команд. Також є й інші функції, такі як документація, запити на прийняття змін (pull requests), історія коммітів, інтеграція з безліччю популярних сервісів, email-повідомлення, емодзі, графіки, вкладені списки завдань, система @ згадок, схожа на ту, що в Twitter, і т.д.

Git - це інструмент, що дозволяє реалізувати розподілену систему контролю версій, а GitHub - це сервіс для проектів, що використовують Git.

Основні можливості GitHub

Етапи розробки. За допомогою вкладки commits ви можете вивчити історію розробки проекту: коли і ким були внесені зміни в які файли і рядки, як розвивався проект в цілому. Коммітів ("commit" - фіксувати) - записи змін в репозиторій.

Контроль версій. Git дозволяє створювати кілька різних, паралельних версій проекту - гілок, кожна з яких призначена для різних цілей. Припустимо, у вас є працюючий проект. Основна гілка проекту - master. У ній зберігається остання стабільна версія. Для додавання в проект нової функціональності необхідно створити окрему гілку, при цьому дані зміни ніяк не позначаються на основну версію. Можна перемикатися між гілками, що призведе до зміни файлів локальної версії сховища

Репозиторій або сховище - це каталог, в якому зберігаються файли вашого проекту. Він може бути розташований в сховище GitHub або в локальному сховищі на вашому комп'ютері. Ви можете зберігати файли кодів, зображення, аудіо або все, що пов'язано з проектом, в сховище.

Гілка це копія вашого сховища. Ви можете використовувати гілку, коли хочете зробити розробку ізольовано.

Робота з гілкою не вплине на центральне сховище або інші гілки. Якщо ви зробили цю роботу, ви можете об'єднати свою гілку з іншими гілками і центральним репозиторієм, використовуючи запит на витяг.

GitHub Issues - один з найпопулярніших трекерів помилок в світі.

GitHub Issues надають власникам сховища можливість організовувати, позначати і пов'язувати питання з певними етапами розробки.

Якщо створити завдання на проект іншого користувача, воно залишиться відкритим до тих пір поки ви або хтось інший не вирішить його або власник сховища не закриє його.

Буває так, що відповіді на ваші питання надходять негайно. А буває і таке, що завдання висить місяцями в очікуванні рішення. За цей час ви отримаєте коментарі інших розробників і надалі зможете повернутися до неї, щоб щось виправити або поліпшити код.

Більшості розробників не платять за підтримку їх коду, опублікованого на GitHub, тому вимагати швидких відповідей на всі ваші питання - нерозумно. Але багато open-Corp

проектів публікуються компаніями, які надають підтримку свого комерційного коду. І вони платять своїм розробниками за їх роботу над проектами.

Однією з відмінних рис GitHub є **система зірок**. Щоб висловити свій інтерес до сховища, його потрібно відзначити зіркою. Це можна зробити за допомогою кнопки «Star». Це дозволить вам відслідковувати цікаві проекти і знаходити схожі.

Це також один з найважливіших рейтингових механізмів, оскільки чим більше зірок має репозиторій, тим він популярніший. Тому в результатах пошуку він буде в топових місцях. Великі проекти можуть мати десятки тисяч зірок.

Також в GitHub є трендова сторінка, на якій представлені репозиторії, які отримують найбільшу кількість зірок за певний період часу (за день, тиждень, місяць).

Форкінг сховища означає, що ви створюєте новий проект на основі існуючого сховища. Говорячи простою мовою, розгалуження сховища означає, що ви копіюєте існуючий репозиторій, вносите деякі необхідні зміни, зберігаєте нову версію в якості нового сховища та називаєте це своїм власним проектом.

Це відмінна функція, яка прискорює розробку проекту. Оскільки це абсолютно новий проект, центральне сховище не буде порушено. Якщо "головний" репозиторій оновлений, ви також можете застосувати це оновлення до вашого поточного ФОРКОМ.

Pull Request. Людина може створити свій власний відгалуження ("fork") проекту, внести деякі зміни і потім зробити Pull Request., Щоб ви замірджіли ці зміни.

Чим популярніша проект, тим більше PR він матиме, як наприклад, проект React:

Як тільки людина робить Pull Request, він розглядається основними розробниками проекту. Залежно від кількості і складності змін, які ви внесли в код, розробнику може знадобитися різна кількість часу, щоб переконатися, що ваші зміни сумісні з проектом.

У проекту може бути чіткий графік змін, які розробники хочуть впровадити. Тоді ваші запити на внесення змін в код будуть розглянуті швидко.

Але важливо ще знати, що ваш запит може і не бути розглянутий зовсім.

Управління проектами. Крім зворотного зв'язку і нових знайомств, GitHub також надає деякі функції з управління проектами.

Одна з таких функцій - Projects. Нововведення в GitHub. Вона ґрунтується на методології Канбан, яка допомагає ефективно організувати роботу.

Wiki призначений для використання в якості документації для користувачів. Одним з найбільш вражаючих видів використання Wiki, є мова програмування Go GitHub Wiki.

Ще один популярний інструмент для управління проектами - віхи. Це частина сторінки "issues". Використовуючи цю функцію, ви можете призначати контрольні точки проекту.

Теги в Git - це показник на конкретний Комміт, і якщо він виконується послідовно, він допомагає вам повернутися до попередньої версії вашого коду, не посилаючись на конкретні коммітів.

Порівняння коммітів. GitHub пропонує безліч інструментів для роботи з вашим кодом. Одна з найбільш важливих речей, які ви можете зробити - порівняти одну гілку з іншого. Або ви можете порівняти останній Комміт з використовуваною вами версією, щоб побачити, які зміни були внесені.

Ключові особливості GitHub:

- Спільна робота з управлінням великими командами.
- Графічне представлення.
- Особисті сторінки проектів.
- Вікі.
- Система відстеження помилок.
- Перегляд файлів проектів.
- Підсвічування синтаксису для більшості мов.
- Приватний репозиторій.
- Пряме додавання нових файлів в репозиторій.
- Код проектів можна скопіювати через Git або завантажити.
- Підтримка отримання і редагування через SVN і Mercurial.
- Pastebin-сепіс gist.github.com для миттєвої публікації фрагментів.

- Вбудоване відстеження завдань і помилок.
- Фільтри, призначення і мітки у завдань.
- Коментарі, сортування і час оновлення.
- Сполучення клавіш.
- Управління віхами.
- Управління доступом.
- Markdown-розмітка.
- Додавання зображень.
- Підтримка SSL, HTTPS і SSH.

Вартість. Безкоштовно - базові можливості GitHub для кожного розробника

Pro - \$ 7 на місяць - професійні інструменти для розробників з розширеними вимогами

Team - \$ 9 (1 користувача / місяць) - розширена співпраця та інструменти управління для команд

Час потрібний на опанування. Щоб зрозуміти основи Git, знадобиться 15 хвилин. try.github.io (<https://try.github.io/>) - крутий інтерактивний туторіал, який просувається самим GitHub, де ви можете дізнатися все основи (безкоштовно!).

GitHub також пропонує безкоштовне навчання та додаткові навчальні ресурси в своїй документації (<https://help.github.com/articles/git-and-github-learning-resources/>).

Просто вивчайте Git. Ви не пошкодуєте про це. Через 15 хвилин ви вже будете розуміти основи, а через кілька годин ви вже зможете зробити свій перший внесок в open source проекти

Платформи та синхронізація. Через WEB-додаток на всіх платформах

Інтеграція з іншими інструментами. Має тісну інтеграцію з засобами розробки Microsoft Visual Studio та всіх рішеннях компанії JetBrains.

Галузь використання. Віддалений контроль версій може використовуватися в усіх сферах де є документообіг. Зараз найпоширенішою ланкою є сфера ІТ, використання систем контролю версій в цій галузі є гарним тоном та методом збереження нервів.

Висновок. GitHub - це чудовий інструмент, який допоможе зробити процес написання коду ще простіше і приємніше. Більш того, це відмінна можливість працювати над реальними проектами. GitHub - це те, що буде однаково корисним як для початківців програмістів, так і досвідчених. Матеріал підготовлений на підставі особистого досвіду автора, а також з використанням джерел [1, 2, 3].

ЛІТЕРАТУРА

1. <https://tproger.ru/translations/difference-between-git-and-github/amp/> - Git і GitHub: що це таке і в чому різниця
2. <https://www.hostinger.ru/rukovodstva/chto-takoe-github/amp/> - Що Таке GitHub І Для Чого Саме Він потрібен?
3. <https://vc.ru/amp/45035> - Знайомство з GitHub

Сітко Денис студент, бакалаврату;
Юрчук І.А., к.ф.-м.н., доцент кафедри
 кафедри Програмних систем і технологій
 Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Microsoft Project програмне забезпечення для управління проектами

Якісний менеджмент проектів є важливою складовою успіху підприємства. Важко керувати великим проектом, оперуючи лише доброю пам'яттю. Тому було створено доволі багато різних програмних продуктів для оптимізації процесів управління проектами.

Microsoft Project – програмне забезпечення від Microsoft для управління проектами. Microsoft Project створений задля оптимізації процесів менеджменту проектів, а саме допомагає систематизувати розробці планів, розподіл ресурсів відповідно до задач, полегшує відстеження прогресу і аналіз обсягів робіт [1, 2, 3].

MS Project зовні є дещо схожим до MS Excel, оскільки формалізація проекту виглядає як таблиця із певними стовпцями та полями. Стовпці можна додавати, видаляти. При додаванні стовпців MS Project пропонує на вибір велику кількість готових типів стовпців, але можна і самому надавати їм ім'я.

Незважаючи на зовнішню складність, MS Project є доволі простим в ідейному плані. Він оперує трьома сутностями – завдання, ресурси, календар і зв'язку між ними. По суті - це база даних, користувацький інтерфейс для створення і редагування сутностей, а також досить проста автоматизація, вбудована в бізнес-логіку даного ПО. Календар та зв'язки між сутностями вбудовуються через функціональний блок «Проект». Для моніторингу певних процесів проекту та аналізу прогресу MS Project існують вбудовані функції, що є доволі простими у використанні. На основі даних щодо задач та ресурсів Project вміє робити різні уявлення з використанням фільтрів, угруповань, сортувань та інших специфікацій. Крім цього він надає можливість за певним алгоритмом обчислювати терміни початку і закінчення завдань з урахуванням доступності призначених ресурсів і зв'язків між завданнями.

Вище описані методи візуалізації більше підходять для проміжних результатів. Таким чином в MS Project існує спеціальне меню «Звіт». Для створення звіту можна конкретизувати велику кількість критеріїв, тобто використовувати фільтри, угруповання, сортування та створювати певні зв'язки між звітними даними.

Вартість. Існує два сценарії роботи з MS Project: за щомісячною підпискою у хмарі (веб-додаток) та покупка локального ліцензійного продукту.

Щомісячні хмарні підписки існують трьох типів:

- 10\$/місяць Project Plan 1 – дана підписка є базовою, надає всі необхідні інструменти для планування проектів локально.
- 30\$/місяць Project Online – дана підписка є середньою, надає всі необхідні інструменти для планування проектів як локально так і в мережі завдяки додатку Project Server.
- 55\$/місяць Project Online Premium – дана підписка є максимальною, надає усі функції підписки Project Online та корисні функції для керування ресурсами в Big Enterprise.

Локальне рішення:

- 20999 грн за Project Standard.
- 40999 грн за Project Pro.

Час потрібний на опанування. MS Project є одним із лідерів ринку в своїй категорії, тому існує велика кількість матеріалів у мережі. До того ж Microsoft має певний уніфікований інтерфейс для офісних програм, таким чином задачі низького та середнього рівня можна виконати без документації до ПО при умові, що ви раніше використовували MS Office (Word, Excel, PowerPoint).

Таким чином опанування MS Project не потребує великих зусиль, а матеріалів та курсів щодо використання існує широкий вибір. Більше того, в програмних продуктах від Microsoft існують підказки при наведенні курсора миші на елемент керування.

Платформи та синхронізація. Існують дві платформи для використання даного ПЗ: веб-додаток та локальне рішення для ПК. Синхронізація даних забезпечується через аккаунт Microsoft чи опціонально через Microsoft Server.

Інтеграція з іншими інструментами. В MS Project існують такі вбудовані інтеграції:

- із поштовим клієнтом на вибір (бажано MS Outlook) для створення розсилок.
- із SharePoint – ПО для організації командної роботи.
- із Team Foundation Server.
- можливість відкривати Excel-документи.

Можливість експорту файлів у формати PDF/XPS.

Для інтеграції MS Project з іншими програмними продуктами, такими як Jira чи Redmine існують плагіни, що можна завантажити в мережі інтернет.

Висновок. Microsoft Project – програмний продукт для керування проектами, є одним із лідерів ринку. Популярність даного продукту не в останню чергу досягається через наявність певної екосистеми з програмних продуктів від Microsoft – від Windows OS до MS Office та інших продуктів.

MS Project надає можливості, які допоможуть організації отримати більш досконалий контроль і видимість протягом проекту, оскільки надає можливість легко систематизувати бізнес-процеси в простій читабельній формі, а великий вибір візуальних уявлень даних та вбудованого створення звітів допомагає акцентовано переглядати лише потрібну та актуальну інформацію на певний момент часу.

Видимість прогресу проекту надає можливість приймати зважені рішення щодо майбутнього. У свою чергу це збільшує операційну ефективність, так як Project надає всі необхідні дані в потрібній формі.

Project допомагає спланувати роботу і ефективно управляти ресурсами. Він також дозволяє ефективно спілкуватися та співпрацювати. До того ж він допомагає надати швидкий доступ до інформації колегам по проекту.

Текст написаний на підставі особистого досвіду автора, а також з використанням таких джерел [1, 2, 3]

ЛІТЕРАТУРА

1. <https://products.office.com/uk-ua/project/project-management-software?rtc=1> - офіційний сайт Майкрософт
2. <https://habr.com/ru/post/151593/> - Використання MS Project для управління проектами з розробки ПЗ
3. https://ru.wikipedia.org/wiki/Microsoft_Project - Стаття про MS Project у Вікіпедії

Дронь Дмитро, студент бакалаврату,
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна,
 співробітник компанії **Stearling Web Agency**;
Меркулова К.В., к.т.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Worksection - сервіс управління проектами для бізнесу

Worksection - це сервіс управління проектами для бізнесу. Дуже добре підходить для зростаючих компаній, в яких необхідно систематизувати свої завдання. Особливо для крос-команд, де є як інженери, так і "звичайні" люди [1, 2, 3, 4]. Першочергово Worksection слід розглядати, як систему керування проектами. У Worksection наступна структура проектів і задач:

- Компанії
- Проекти
- Задачі
- Підзадачі рівень 1
- Підзадачі рівень 2

Проекти групуються по компаніям, до яких вони належать. У кожній задачі та підзадачі існують чеклісти. У них ви можете писати зовсім прості завдання, які не потребують термінів або окремого делегування. Основними одиницями сервісу Worksection є проекти та завдання. Створити новий проект дуже просто. Перебуваючи на будь-якій сторінці, наприклад, на Головній сторінці або на сторінці «Проекти» натисніть на зелений плюсик і на пункт **«Новий проект»**[3]. У вікні введіть всі необхідні дані і підтвердьте створення нового проекту. Якщо якісь дані вам невідомі, ви легко можете їх додати або змінити пізніше.

Сторінка створення/редагування проекту виглядає так: У верхній частині розташовані налаштування назви, опису, строків і передбачуваних витрат. У нижній — компанії, що беруть участь у проекті, та їх працівники, які отримали доступ до проекту (вони відзначені галочками).

Нерідко виникає потреба перемкнутись на інший проект, для постановки нових завдань, формування звітності і т.д. Для цього є розділ «Проекти». Учасники проектів можуть бачити всі свої проекти в цьому розділі. Бачать дату старту та дату завершення проекту, відсоток виконаних задач і витрачений час. Крім цього, адміністратори і власник акаунта мають доступ до прихованої від всіх інших людей колонки «Примітки». Зазвичай, у ролі адміністраторів і власника акаунта, виступають директори компаній та їх заступники, у цих колонках вони можуть записувати конфіденційну інформацію, яка допоможе їм правильно розставити пріоритети у виконанні проектів.

Проекти не існують без конкретних задач. Для створення задачі, знаходячись на одній із сторінок проекту, натисніть на зелений плюс і виберіть «Нова задача». Відкривається сторінка створення задачі. На сторінці ви можете додати наступну інформацію:

1. назва задачі
2. вибрати пріоритет
3. призначити відповідального по задачі (можна залишити задачу без відповідального)
4. призначити мітки і статуси задачі
5. встановити строки (старт-фініш або дедлайн) задачі
6. внести передбачувані витрати часу та фінансів)
7. ввести детальний опис задачі
8. прикріпити до задачі файли
9. підписати/відписати на задачу інших співробітників, а також обмежити видимість задачі (натиснувши на «замок»)

Також ви можете створювати відразу групу задач, а завдяки спеціальним стрілкам — змінювати ієрархію задач. Іншими словами, за один раз ви можете додати як задачі, так і вкладені в них підзадачі, і навіть — під-підзадачі.

Однією з вагомих переваг сервісу Worksection є звітність, адже контроль фактичних витрат відносно до запланованих є невід'ємною частиною кожного успішного проекту чи компанії. Спеціально для таких потреб у розділі **Звіт** на сторінці **«Детальний»** відображені всі задачі і підзадачі, у яких були внесені витрати за вказаний період. А також всі задачі і підзадачі, закриті за цей період.

Статистика учасників (сторінка «По людям») допоможе вам оцінити ефективність роботи ваших співробітників. Режим по днях — подивитися, хто і скільки часу витратив на роботу кожен день. А статистика «За проектами» — розставити пріоритети.

Для комфорту користувача передбачена можливість експорту звітності в excel-таблицю. Така гнучкість у виборі параметрів для заповнення даних дозволяє максимально конкретизувати робочий процес, і зробити його прозорим.

Вартість. Додаток розповсюджується методом платної підписки.[4] Спробувати можна безкоштовно, доступний функціонал буде достатньо обмеженим. Попри це існує можливість безкоштовного користування за тарифом «Базовий».

Доступні такі тарифи:

1. Free – \$0 – містить обмежений функціонал, якого достатньо у випадку невеликої команди.
2. Міні- \$29/місяць – містить всі необхідні інструменти для бізнесу, проте кількість користувачів не має перевищувати 10 чоловік.
3. Базовий- \$49/місяць – функціонально аналогічний до попереднього тарифу, проте має більшу кількість доступних користувачів та доступного місця у сховищі даних, що буде комфортно для невеликої компанії.
4. Бізнес- \$99/місяць – для великих організацій.
5. Преміум- \$199/місяць – для великих організацій, які турбуються про свою безпеку, адже сервіс надає сервер та домен для користування компанією.
6. Enterprise - по запиту – для великих організацій. Даний тариф не має обмежень у користуванні ресурсами сервісу.

Час потрібний на опанування. Worksection має інтуїтивно зрозумілий інтерфейс, тому почати роботу можна без жодних перешкод. У випадку виникнення питань, щодо користування тою чи іншою функцією на сайті сервісу є розділ “FAQ”, який дасть відповідь на будь яке питання, що може виникнути.

Платформи та синхронізація. З сервісом можна працювати через звичайний браузер. Окрім цього є додатки для Android та IOS.

Вся інформація за вашого аккаунту синхронізується та є доступною в будь-яких момент часу з будь-якої платформи.

Інтеграція з іншими інструментами. Є можливість інтеграції з

- Telegram Messenger
- Google Диск
- Google Документи
- Google Календар
- Slack

Також у Worksection є відкритий API для взаємодії з проектами. За допомогою API Worksection дозволяє вам отримати доступ до сервісу та запросити/відіслати дані з метою синхронної взаємодії сервісу з іншими програмами.

Є найнеобхідніші функції, які допоможуть Вам коректно вести свою діяльність, а саме:

- отримання списку проектів, завдань, підзадач, а також списку тимчасових і фінансових витрат;
- створення і закриття завдань, підзадач, розміщення коментарів;
- підписка і відписка учасників;
- можливість застосування файлів.

Рекомендована галузь використання. Worksection підійде як для персонального, так і командного використання. Основними напрямками використання є:

1. Планування проектів
2. Контроль термінів і завдань
3. Робота з клієнтами та фрілансерами
4. Ведення проектного бюджетування - план / факт звітів.
5. Формування звітів про виконану роботу, хто скільки зробив і скільки часу витрачено
6. Вважати собівартість проектів і команди
7. Візуалізація роботи за допомогою дощок для задач (канбан-борди)
8. Ведення календаря та діаграми Ганта

Висновок. Worksection є майже безкомпромісним засобом в сфері керування проектами. Даний сервіс має весь потрібний функціонал для цього, проте порівняно з вагомими конкурентами в обличчях сервісів Jira та Asana, має деякі недоліки, а саме:

- не зручний мобільний додаток
- відсутність можливості управління версіями

В цілому порівняно з конкурентами, Worksection більш направлений на компанії малого та середнього розміру масштабованості, проте використовувати даний сервіс можна при будь-яких умовах масштабованості. Даний продукт стрімко розвивається і в майбутньому може скласти конкуренцію на більш високому рівні, ніж зараз.

Текст написаний на підставі особистого досвіду автора, а також з використанням літератури [1, 2, 3, 4].

ЛІТЕРАТУРА

1. Сайт компанії <https://worksection.com/ua/>
2. <https://срм.укр/catalog-crm/cloud/worksection>
3. <https://worksection.com/faq/overview.html> - огляд worksection
4. <https://worksection.com/ua/price.html> - огляд цін worksection

Цехмейструк Андрій, студент бакалаврату,
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна,
співробітник компанії **УКРСІББАНК**;

Доценко С.І., к.ф.-м.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

JIRA - засіб відстеження завдань та управління проектами

Jira - комерційна система відслідковування завдань, помилок, призначена для організації взаємодії з користувачами. В деяких випадках використовується і для управління проектами [1, 2, 3, 4, 5]. Має веб-інтерфейс. Розроблена австралійською компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Перший випуск JIRA стався в 2002 році. Назва «JIRA» виглядає як аббревіатура, але це не так. Насправді, JIRA - усічений варіант Gojira, японського вимови для Godzilla. Фактично, це відсилання на Bugzilla, головного конкурента JIRA.

JIRA складається з ряду **компонентів**, кожен з яких можна налаштувати:

- Робочий процес (Workflow);
- Типи завдань (Issue Types);
- Призначені для користувача робочі простори (Custom Fields);
- Вікна (Screens);
- Налаштування робочих просторів (Field Configuration);
- Повідомлення (Notification);
- Рішення (Permissions).

Реалізація. Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах. Основний елемент обліку в системі - завдання (англ. Ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширене додатковими полями (також можуть бути визначені і нові поля користувача), додатками (наприклад - фотографіями, скриншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через налаштування потоку операцій. Будь-які зміни в задачі записуються в журнал.

Можливості. Має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (англ. Screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, управління вимогами, управління ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Концептуальні поняття JIRA: Завдання, Проект і Процес.

Завдання JIRA. Фактично, все, що можна створити і відстежувати за допомогою JIRA, можна назвати завданням:

- Баг
- Проектне завдання
- Форма залишеного запиту
- Довідкова картка
- JIRA процес

Процес (Workflow) JIRA - це набір статусів і переходів, через які проходить помилка / завдання / проблема протягом її життєвого циклу. Фактично, він представляє внутрішні процеси вашої організації.

П'ять основних **етапів життєвого циклу завдання**:

- Відкрита задача
- Вирішена задача

- Вирішуєма задача
- Повторна задача
- Закрита задача

Складові частини процесу в JIRA:

- Статуси (Statuses) визначають позиції завдання в поточному процесі.
- Переходи (Transitions) показують перехід завдання з одного статусу в інший.
- Особа на яку призначена задача (An assignee) це особа, відповідальна за

виконання завдання.

- У Резолюціях (Resolutions) міститься пояснення переходу завдання з відкритого стану в закрите.
- Умови (Conditions) контролюють співробітника, який може виконати перехід.
- Валідатори (Validators) перевіряють, чи виконаний вхід до переходу.
- Властивості (Properties) використовуються для реалізації обмежень на певні кроки чи переходи процесу.
- Пост функції (Post functions) дозволяють виконувати будь-яку додаткову обробку, необхідну після виконання переходу, наприклад:
 - Створення історії змін завдання
 - Створення події для відправки повідомлень по електронній пошті
 - Оновлення полів завдання
 - Додавання коментаря до задачі

Існуючі активні і неактивні процеси. Якщо процес в даний час використовується в одному або декількох проектах, це активний процес, в іншому випадку це неактивний процес.

Існує спеціальний графічний інструмент, який допомагає користувачам бачити макет процесу і створювати або редагувати кроки і переходи процесу. Це дизайнер процесу.

Створення процесів

Існує 3 способи запуску нового процесу:

- Створення нового процесу
- Клонування існуючого процесу
- Імпорт процесу

Проект JIRA - це набір завдань. Основними атрибутами проекту є:

- Ім'я (Name) вибирається адміністратором.
- Ключ (Key) - це ідентифікатор, з якого починаються всі назви завдань в рамках проекту.
- Компонент проекту (Project component) - це логічна угруповання завдань в рамках проекту.

Часто необхідно, щоб конкретна проблема була пов'язана з версією проекту. Існує три стану версій: Випущені (Released), невипущеними (Unreleased) або Архівовані (Archived).

Схеми захисту завдань JIRA (Issue Security Schemes) дозволяє контролювати доступ до завдань. Вона включає в себе кілька рівнів доступу, які розподіляються між користувачами і групами. Ви можете вказати рівень доступу до завдання під час її створення або редагування. Також є Стандартна схема захисту (Default Permission Scheme), яка буде призначена будь-якого нового проекту. Схеми захисту дозволяють вам створювати набори рівнів доступу і застосовувати їх до будь-якого проекту.

Системна адміністрація (System Administration)

Ось кілька корисних функцій, які JIRA надає адміністраторам:

- Список ревізій (Audit Log). У цій вкладці ви можете побачити деталі створеної завдання, а також зміни, внесені в завдання.
- Зв'язування завдань (Issue Linking). Тут вказується пов'язана ваша задача з якоїсь іншої, яка існує в даному проекті. Також в цій панелі можна скасувати даний зв'язок.
- Система пошти JIRA (Mail in JIRA). Використовуючи систему пошти в якості адміністратора, ви можете пересилати завдання на поштові сервера POP і IMAP, а також відправляти їх у вигляді повідомлень на зовнішні поштові скриньки.

- Події (Events). У цій вкладці описаний статус, стандартний шаблон, схеми оповіщення і передача відповідальності за подію. Події розділені на два типи: Системні події (System event, ті, що встановлені в JIRA за замовчуванням) і для користувача події (Custom event, відповідно, ті, що були створені користувачами).

- Контрольний список (Watch list). Дозволяє переглядати певні завдання, бачачи повідомлення, пов'язані з ними. Щоб переглянути завдання, клікніть «перегляд» у вікні завдання, а якщо ви хочете побачити, хто ще переглядає цю задачу, ви можете натиснути на число в дужках.

- Лічильник завдань (Issue Collectors). Дозволяє збирати інформацію з будь-якого сайту. Будучи адміністратором, можна клікнути по лічильнику задач, після чого з'явиться опція, що дозволяє його додати. Як тільки ви налаштуєте зовнішній вигляд лічильника, автоматично згенерований JavaScript можна перенести на сайт для передачі інформації.

- Інструменти розробки (Development Tools). Ви можете також підключити ваші інструменти розробки ПО до JIRA, використовуючи функції адміністратора. Вам необхідно ввести URL додатки для підключення його до JIRA.

Пакети JIRA. В даний час основними пакетами JIRA є:

- JIRA Core
- JIRA Software
- JIRA Service Desk

Основним пакетом є JIRA Core. Це оптимальне поєднання потужних можливостей для спільної роботи з завданнями в процесі або проект.

Ви можете керувати різними областями і напрямками бізнесу з JIRA Core:

- Фінанси
- Маркетинг
- Внутрішні операції
- Юридична документація
- HR

JIRA Software - це інструмент управління проектами для Agile команд. Простіше кажучи, JIRA Software - це JIRA Core плюс плагін JIRA Agile.

Доповнення JIRA. Якщо ви хочете використовувати весь потенціал JIRA, ви також повинні використовувати її доповнення. На сьогоднішній день Atlassian пропонує понад 2600 доповнень. Розглянемо деякі з них.

Zephyr for JIRA дозволяє користувачам організовувати, керувати, шукати, створювати, планувати, виконувати і повідомляти про дії з тестування проектів, які активно управляються в JIRA. Основні **особливості і переваги** Zephyr:

- Тестування аналогічне JIRA. Завданнями можна управляти, як і з будь-якої іншої завданням JIRA.

- Повністю вбудований в JIRA, що дозволяє легко почати тестування.

- Потужні можливості пошуку з використанням Zephyr Query Language і Execution Navigator.

- Повний набір функцій управління тест кейсами.

- Тісна інтеграція з іншими системами JIRA.

- Індивідуальні панелі і показники з індивідуальною конфігурацією.

- Тестові приклади можна експортувати у формати word, excel.

- Можна створити кілька тестових ітерацій з декількома циклами.

- Краща простежуваності з прив'язкою між історіями, тестовими прикладами і помилками в циклах тестування.

- Процес розгляду тест кейса можливий через Agile board.

- Індивідуальні фільтри.

Обмеження Zephyr:

- Неможливо імпортувати результат тесту.
- Неможливо експортувати всі тест кейси та результати їх виконання в один файл.
- Імпортувати існуючі тестові приклади з excel не просто, і для цього потрібно використовувати деякі зовнішні настройки.
- Звітність може бути виконана тільки для одного конкретного циклу ітерації.
- Накопичувальний звіт не може бути створений для всіх циклів тестування ітерації.
- Тестові приклади мають лише 3 стандартних стовпця. Неможливо налаштувати формат тестового сценарію.
- Інтеграція пакетів автоматизації може бути виконана за допомогою Selenium RC.
- Інтеграція з Selenium Web Driver не реалізована.

Adaptavist ScriptRunner for JIRA є вбудованим редактором для написання відмінних сценаріїв. Особливості Adaptavist ScriptRunner:

- Розширення функціональності JIRA
- Поліпшення призначених для користувача звітів з розширеними функціями JQL (hasLinks, hasComments, hasAttachments або subTasksof)

- Обробники подій JIRA
- Автоматична дія з зазначеними запитами JQL

JIRA Toolkit Plugin - це унікальний набір розширень JIRA. Деякі з його можливостей:

- Містить велику колекцію корисних полів, що настраюються
- Плагін JIRA Toolkit допомагає фільтрувати інформацію, яка в іншому випадку була б недоступна
- Надає простий спосіб обходу обмежень JIRA або рішень деяких конкретних проблем.

JIRA Misc Workflow Extensions використовується для реалізації складних процесів в JIRA Server і JIRA Cloud. Можливості JIRA Misc Workflow Extensions:

- Валідатори процесу
- Умови процесу
- Постобробки процесу
- Тісна інтеграція з іншими продуктами Atlassian
- Проекти JIRA і адміністрування хост-додатків

JIRA Charting Plugin надає різні графіки і звіти для JIRA. Його основні функції:

- Надає додаткові можливості для візуального представлення проекту в різних контекстах.
- Плагін JIRA Charting включає безліч гаджетів і звітів.
- Вистава «Charting» для навігатора завдань.
- Дозволяє отримувати додаткові поля, що настраюються (Дата дозволу, Дата першого відповіді і Час перебування в даному статусі).

Portfolio for JIRA Портфоліо для JIRA - це повномасштабний інструмент управління портфоліо Agile. Його головні **особливості**:

- Повна інтеграція з іншими інструментами Atlassian
- Інструменти для огляду діяльності команди і проекту
- Дозволяє експортувати звіти в CSV і ділитися ними з посиланням
- Він дозволяє створювати власні або міжпроектної версії
- Конфігурація дат випуску

- Дозволяє присвоювати завдання спринті
- Прогрес і відстеження статусу завдання
- Повторне визначення пріоритетів робочих елементів
- Створення декількох сценаріїв типу «що-якщо»
- Планування в історическої і времни контекстах

Tempo Timesheets - це спеціальний інструмент для тайм-трекінгу, інтегрований з JIRA. Основні можливості Tempo Timesheets:

- Допомагає відслідковувати час для обліку, виставлення рахунків клієнтам, підвищення ефективності, розрахунку заробітної плати і прогнозування.
- Tempo Timesheets мінімізує труднощі і надає більш точні дані в режимі реального часу.
- Допомагає управляти витратами вашого бізнесу.
- Tempo Timesheets надає безліч інструментів внутрішнього управління часом.
- Він пропонує гнучкі звіти про організаційні ресурси і діях.

Вартість. Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатку. Ціноутворення залежить від максимального числа користувачів, при цьому близько \$ 50 за користувача для локального і \$ 7 на місяць за користувача для віддаленого доступу є типовими цінами.

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

- проект використовує ліцензії, схвалені Open Source Initiative;
- вихідний код проекту доступний для скачування;
- у проекту є публічно доступна веб-сайт;
- програмне забезпечення від Atlassian є на веб-сайті проекту.

Час потрібний на опанування. Jira має дуже привабливий та зручний інтерфейс. Щоб розібратися в додатку знадобиться менше години, проте для повного вивчення функціоналу системи може пройти декілька місяців, система має безліч доповнень та ще більше варіантів налаштування. Кожен користувач може створити власний пакет налаштувань для покращення роботи.

Платформи та синхронізація. Доступний на усіх платформах, а також має WEB-додаток.

Інтеграція з іншими інструментами. Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce. Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи російську, англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи - плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Рекомендована галузь використання. В управлінні, розробці та адмініструванні програмного забезпечення. Це потужний інструмент для колаборації різних сфер діяльності в IT та інших індустріях. Даний продукт допомагає організувати робочий процес та вивести його на новий систематичний рівень з збільшеною продуктивністю.

Висновки. Плюси:

- Функції, які недоступні в будь-якому іншому інструменті.
- Користувачі отримують останні оновлення про хід виконання проектів.

- Яскрава і своєчасна політика ліцензування.
- Її можна запустити будь-де.
- Багато відомих компаній і установи використовують JIRA (Nokia, LG, Європейська комісія, Європейський парламент, Гарвардський і Стенфордський університети і т.д.)
 - Легко налаштовується і розширюється.
 - Інтерфейси JIRA прості і зручні.
 - Розроблена на основі сучасних веб-технологій.
 - Світова спільнота JIRA. Багато компаній по всьому світу постійно розробляють унікальні розширення і модулі.
 - Легко інтегрується з найпопулярнішими інструментами.
 - Має велике розмаїття доповнень, що робить її найпотужнішим інструментом для розробки, управління проектами та управління тестуванням.

Висновки. Мінуси:

- Важке налаштування.
- Потрібно багато часу, щоб навчитися ефективно використовувати JIRA.
- Бракує багато функцій. Ця проблема може бути вирішена за допомогою різних плагінів і налаштувань, що іноді занадто дорого.
- Мобільний додаток JIRA має обмежену функціональність.
- Занадто складна для малого бізнесу і невеликих команд.

Матеріал підготовлений на основі особистого досвіду автора з використанням джерел [1, 2, 3, 4, 5].

ЛІТЕРАТУРА

1. <https://startpack.ru/application/jira> - Стаття про Jira на сайті [startpack](https://startpack.ru)
2. <https://bytextest.ru/2018/08/31/jira-dlya-novichkov/> - JIRA: докладне керівництво для новачків
3. <https://geteasyqa.com/ru/blog/jira-test-case-project-management-tool/> - JIRA - інструмент для тестування і управління проектами
4. <http://jira.ru/jira.shtml> - Система управління проектами і завданнями JIRA компанії Atlassian і її застосування
5. <https://softlist.com.ua/articles/chto-takoe-jira/> - Що таке Jira?

Панасовський Дмитро, студент бакалаврату;
Ковалюк Т.В., к.т.н., доцент кафедри
 кафедра Програмних систем і технологій
 Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Notion - єдиний інструмент для ведення особистих та корпоративних справ і задач

Notion – це додаток, який позиціонується як єдиний інструмент для ведення особистих та корпоративних справ і задач [1, 2, 3]. Якщо коротко то це:

1. Створення заміток та списків задач
2. Ведення корпоративної Вікі
3. Управління проектами
4. Створення персональних баз даних

Цей сервіс об'єднує в собі величезну кількість інструментів. Нотатки та списки справ, документи та таблиці, канбан-дошки і бази знань - тут є все, з чим нам доводиться мати справу щодня. Сервіс являє все це у вигляді єдиного робочого простору. Подібно деталей LEGO, в нього легко додати потрібні елементи і зібрати свій ідеальний інструмент продуктивності для зберігання ідей, планування і спільної роботи з колегами. Notion дозволяє вести всі процеси в чистому, універсальному середовищі і тримати їх під контролем.

Перш за все в Notion створюється *робоча область*, яка представляє собою деревовидну структуру довільної глибини, яка складається із *сторінок*. У свою чергу сторінка складається з *блоків*.

Основною одиницею усіх сторінок в Notion є блок. *Блок* – це будь-яких фрагмент вмісту сторінки таких як наприклад абзац тексту, пронумерований список, рядок таблиці, відео тощо. Це досить простий спосіб будувати власні сторінки «цеглинка за цеглинкою».

Серед основних елементів, які можна додати на сторінки такі:

- Текст
- Посилання на іншу сторінку
- To-do список
- Заголовок
- Пронумерований та пронумерований списки
- Цитата
- Посилання
- Дата
- Таблиця / дошка / календар / лист / галерея
- Зображення
- Медіафайл
- Код
- Файл
- Формула
- Та вбудовані елементи для зовнішніх сервісів (twitter, youtube, maps)

Основною перевагою Notion є можливість додавання на сторінку таблиці бази даних з повною її кастомізацією. Таблиця представляє собою довільну кількість рядків та колонок, які можуть бути формату тексту, числом, датою, полем з мультिवибором, галкою, посиланням тощо. Після створення таблиці та наповнення її даними, є можливість створити один чи декілька видів відображення цих даних, серед яких

- Таблиця
- Календар
- Дошка
- Галерея
- Список

В новому виді можна змінити та відфільтрувати дані, згрупувати їх по будь-якому полю, приєднати інші таблиці. Саме завдяки такому гнучкому налаштуванню таблиць і

можливості їх поширення між членами команди, можна створити сторінки майже для будь-чого. Це ведення канбан дошки, ведення бюджету, управління задачами, списки літератури, управління знаннями, дорожні карти, Вікі продукту, скрам планування спринтів та багато іншого.

Вартість. Додаток розповсюджується методом платної підписки. Спробувати можна безкоштовно, але є обмеження на об'єм завантажуваних даних. При звичайному користуванні швидко досягається ліміт безкоштовної версії. Доступні варіанти підписок:

1. Free – \$0 – 1000 ліміту на кількість блоків
2. Personal - \$4/місяць – ідеально для персонального користування, адже відкриває всі можливості додатку без лімітів окрім користування в команді.
3. Team - \$8/місяць (кожен з команди) – підходить для проектних команд невеликого розміру.
4. Enterprise - \$20/місяць (кожен з команди) – для великих організацій.

Час потрібний на опанування. Notion має інтуїтивно зрозумілий інтерфейс, тому створити першу сторінку буде дуже просто. Основною складністю для багатьох буде правильно організувати свій робочий простір. На сайті розробника є багато навчальних матеріалів, а також шаблонів сторінок, тому після кількох днів використання Notion стане невід'ємною частиною вашого робочого простору.

Також є великий форум із купою матеріалів, що відповідають на більшість питань стосовно використання додатку. Для тих, хто не знайшов відповіді, розробники відповідають протягом одного робочого дня в спеціальному чаті.

Платформи та синхронізація. З сервісом можна працювати через звичайний браузер або окремо через встановлений додаток для macOS або Windows. Окрім цього є додатки для Android та IOS. Вся інформація з вашого аккаунту синхронізується та є доступною в будь-яких момент часу з будь-якої платформи.

Інтеграція з іншими інструментами. В Notion є можливість імпорту інформації з

- Evernote
- Google Docs
- Confluence
- Trello
- Asana
- Quip
- Dropbox
- Hackpad
- Word and Excel
- WorkFlowy
- CSV

Окрім цього є можливість експорту ваших даних в pdf, html, CSV.

Для професійного використання є відкритий API для взаємодії з проектами. Є можливість створення, заповнення та перегляду ваших баз даних. Тому під Notion можливо написати власний додаток, що буде спрощувати та автоматизувати роботу з ним, якщо буде така необхідність.

Рекомендована галузь використання. Notion підійде як для персонального, так і командного використання. Основними напрямками використання є:

1. Створення заміток та списків задач, їх поширення в команді
2. Ведення корпоративної Вікі для ваших проектів
3. Управління проектами, назначайте пріоритети та терміни виконання.
4. Створення персональних баз даних, повна кастомізація. Адже важливо не тільки добавляти інформацію, а й правильно і інформативно відображати її.

Висновок. На даний момент Notion не має рівних собі аналогів для ведення персональних баз даних та заміток. Для всіх, кому важливо тримати всю інформацію в одному місці він стане незамінним помічником за розумну ціну. Для використання Notion як інструменту для командної розробки, необхідне правильне налаштування сторінок і таблиць, а також придбання платної підписки кожному члену команди.

Серед недоліків можна виділити неможливість безкоштовного використання при роботі в команді, досить повільний мобільний додаток, відсутність управління версіями та відсутність української/російської мов.

Текст написаний на підставі особистого досвіду автора, а також з використанням посилань на наступну літературу [1, 2, 3]

ЛІТЕРАТУРА

1. <https://www.notion.so/about> - Історія про засоби та майбутнє праці
2. <https://medium.com/mactools/notion-работа-над-ошибками-2905dc99b8cd> - Поняття. Використання і налаштування під себе
3. <https://lifel hacker.ru/prilozhenie-notion/> - Обзор понятия - универсальный инструментальный продукт для работы и життя

Шаламай Єлизавета, студент бакалаврату,
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна
 співробітник компанії **Eram Systems**;
Поляков С.А., к.ф.-м.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Microsoft Teams - уніфікована платформа для спілкування та співпраці

Microsoft Teams - це уніфікована платформа для спілкування та співпраці, яка поєднує чати, відео зустрічі, зберігання файлів (включаючи спільний доступ до файлів) та інтеграцію з іншими програмами. Цей сервіс інтегрується з пакетом Office 365 за спеціальною підпискою та пропонує розширення, які можуть інтегруватися з продуктами, що не належать Microsoft.

Microsoft Teams — центр для командної роботи в Office 365 від Microsoft, який інтегрує користувачів, вміст і засоби, необхідні команді для ефективнішої роботи. Застосунок об'єднує все в спільному робочому середовищі, яке містить чат для нарад, файлообмінник та корпоративні програми. Розроблений для смартфонів, що працюють на платформах Android, iOS, Windows Phone і комп'ютерів з операційною системою Windows або Mac OS. Microsoft Teams є конкурентом таких сервісів, як Slack, і є еволюційним оновленням від Microsoft Skype для бізнесу [1]. Microsoft Teams доступні клієнтам Office 365 по всьому світу на 53 мовах.

Функціональність

Команда – це колекція людей, розмов, файлів і інструментів – все це в одному розташуванні.

Канал – це місце для командних обговорень, присвячених підрозділу, проекту або темі [2].

Зліва на панелі меню розташована вкладка **Команди (Teams)**. Команди складаються з **каналів (channels)**. Кожен канал може представляти тему, відділ, або за будь-якою іншою характеристикою. Саме за допомогою каналів можна організовувати командні наради та **зустрічі (meetings)**, обмінюватися повідомленнями та ділитися файлами. Для зручності навігації вгорі в кожному каналі є вкладки-посилання на сховища файлів, інтегровані додатки та сервіси, тощо.

Для командних нарад є можливість презентувати екран, вести діалог в спеціальному чаті зустрічі, або записувати зустріч на відео для подальшого перегляду.

Файли, що додаються до каналу, знаходяться у вільному доступі для всієї команди та можуть переглядатися та редагуватися усіма учасниками. Список усіх доступних файлів можна знайти, перейшовши на вкладку **Файли (Files)** в меню навігації вгорі. Також на боковій панелі є окремий пункт Файли, який надає доступ до файлів з усіх каналів команди.

Окрім каналів, в Teams є можливість створити приватний чат з однією або кількома особами, ввівши відповідні імена членів команди у рядку пошуку. Для зручності чату можна присвоїти назву. Як і для каналу, існує можливість відео- та аудіо- дзвінків з кожного чату (тільки для членів чату).

Вкладка **Зустрічі (Meetings)** представляє собою календар з усіма запланованими нарадами та дзвінками на день або на цілий тиждень. Підключитися можна за допомогою кнопки **Приєднатися (Join)**. Ця вкладка дозволяє також призначити власну зустріч. Цей календар синхронізується з календарем додатку Outlook, що забезпечує ще більш зручну організацію робочого процесу.

Переглянути усі нещодавні події команди - непрочитані повідомлення, відповіді у переписках, згадки та інші - допоможе вкладка **Активності (Activities)**.

Створення та налаштування команди. Щоб організувати роботу в Microsoft Teams, створіть команду, а потім додайте учасників і канали.

- Створення команди
 - Виберіть **Команди > Приєднатися або створити команду**. На цій сторінці ви можете створити власну команду або приєднатися до наявних.
 - Натисніть кнопку **Створити нову команду**, а потім виберіть команду створити **Групу з нуля** або виберіть команду **Створити з...**, щоб створити нову команду або створити з існуючої групи відповідно.
 - Виберіть варіант **Приватна**, якщо ви хочете, щоб користувачі могли запросити дозвіл на приєднання або **Відкрита**, щоб будь-хто в організації міг приєднатися.
 - Надайте команді ім'я та додайте короткий опис (за бажанням).
 - Натисніть **Створити**.
 - Щоб додати учасників за межами організації, запросіть їх як гостей за допомогою адрес електронної пошти. Виберіть **Додати**, а потім – **Закрити**.

- Створення каналу

За замовчуванням кожна команда отримує загальний канал, у який можна публікувати оголошення та інформацію, яка потрібна всій команді. Щоб додати інші канали, виконайте наведені нижче дії.

- Виберіть **Додаткові настройки...** поруч із назвою команди.
- Виберіть **Додати канал**.
- Введіть назву та опис каналу. Ви можете створити канал, присвячений певній темі, проекту, відділу або будь-чому іншому.
- Установіть прапорець **Автоматично відображати цей канал у списку каналів всіх користувачів**, якщо потрібно, щоб цей канал був автоматично видимий у списку каналів усіх користувачів. Натисніть кнопку **Додати**.

- До вподобань каналу виберіть опцію **Улюблені** ☆ поруч з іменем каналу. Це допоможе закріпити канал у списку швидкого доступу і додасть сповіщення про будь-які активності у цьому каналі.

Надання спільного доступу до файлу

Іноді потрібно опублікувати файл для спільного доступу усією командою. Для цього у розмові каналу натисніть кнопку вкласти Кнопка "Вибрати файл" під полем, у якому потрібно ввести повідомлення. Виберіть один із наведених нижче параметрів.

Виберіть файл, > надати спільний доступ до посилання. Якщо файл надсилається з комп'ютера, виберіть файл, натисніть кнопку Відкрити, а потім надішліть Кнопка «Надіслати».

Щоб переглянути всі файли, які ви опублікували в каналі, перейдіть на вкладку **Файли**.

Вартість. Microsoft Teams доступні для клієнтів Office 365 за підпискою на наступні плани:

- Безкоштовно для необмежених повідомлень та пошуку до 300 користувачів із основними функціями
 - 6\$/місяць для користувачів Office 365 Business Essentials for Exchange, Skype для бізнесу та команд Microsoft без настільних програм Office
 - 15\$/місяць на користувача Office 365 Business Premium for Exchange, Skype для бізнесу, команди Microsoft та настільні / мобільні програми Office
- * 5 або 12,50\$/місяць відповідно з річною підпискою

Якщо у вас є підписки вище, Команди Microsoft доступні для вас за замовчуванням [3].

Для користувачів безкоштовної версії на сайті підтримки Microsoft доступний набір навчальних статей, що допоможуть зробити взаємодію через Teams максимально ефективною [4].

Час потрібний на опанування. Загалом Microsoft Teams має доволі невисокий поріг входження, оскільки має дуже зрозумілий і передбачуваний інтерфейс, а також широку інфраструктуру і підтримку з боку Microsoft.

Для першого використання перейдіть до <https://www.office.com> та натисніть Teams.

Для полегшення входження як менеджером, так і розробникам, компанією Microsoft було створено багато навчальних матеріалів, довідка та шаблони використання.

Перейшовши на сторінку dev.office.com/microsoft-teams, ви можете знайти допоміжні матеріали про те, як створювати вкладки, боти та конектори.

Шаблони групи - це готова заготовка визначення структури групи, створена за визначеними потребами або проектом. Шаблони групи можна використовувати для швидкого створення каналів з різних тем та перед встановленням додатків для роботи з важливим контентом і службами [1].

Платформи та синхронізація. Microsoft Teams розроблений для смартфонів, що працюють на платформах Android, iOS, Windows Phone і комп'ютерів з операційною системою Windows або Mac OS. Також представлена веб-версія сервісу [5].

Звичайно, функціональність для різних платформ відрізняється за наповненістю. Наприклад, в веб-версії представлені development tools, що дозволяють доповнювати сервіс власними компонентами, при цьому в настільній версії цей функціонал відсутній. Мобільний додаток взагалі володіє лише базовими функціями, такими як доступ до каналів, чатів, файлів та деякі інші.

Зручним є те, що можна заходити в Microsoft Teams з різних платформ під одним і тим же акаунтом, і зміни будуть автоматично синхронізуватися, якщо є доступ до мережі Інтернет [6].

Інтеграція з іншими інструментами. Для Microsoft Teams є два основні види інтеграції з іншими інструментами: за допомогою вкладок і за допомогою конекторів.

Для будь-якого каналу або чату існує можливість додати на верхню панель вкладку для швидкого доступу до певного сервісу, наприклад Trello, OneNote, Team Foundation Services та безліч інших. Для цього необхідно натиснути на + в правому боці панелі, обрати серед запропонованих варіантів сервіс, що вас цікавить, і здійснити базові налаштування, специфічні для кожного інструмента.

Інша, більш гнучка та наповнена функціональністю опція – це приєднати сервіс за допомогою конектора. Конектори дозволяють користувачам підписатися на отримання сповіщень та повідомлень від ваших веб-служб. Вони виставляють кінцеву точку HTTPS для послуги для розміщення повідомлень (як правило, у вигляді «карток»).

Microsoft надає кілька десятків уже існуючих інструментів, що можуть бути інтегровані таким чином, серед найбільш поширених Github, Jenkins, Visual Studio Team Services, Azure DevOps та багато інших.

Окрім використання стандартних конекторів, є можливість створювати свої власні (більш детально можна почитати на сторінці Microsoft Teams для розробників) [7].

Отже, Microsoft надає надзвичайно широкі можливості для інтеграції та кастомізації Microsoft Teams.

Висновки. Microsoft Teams – це програма для співпраці, яка допомагає вашій команді залишатися організованою та проводити розмови – усе в одному розташуванні. Це універсальність - завантажте Teams для настільного ПК або пристрою iOS чи Android, бо ж використовуйте Teams онлайн на сайті <https://teams.microsoft.com>.

Основні функції Microsoft Teams, які роблять її незамінним сервісом для командної роботи:

Чат для роботи над проектами. Microsoft Teams передбачає організацію чату по бесідам. Групове листування за замовчуванням видно всім, але також є варіант відправки особистих повідомлень.

Хаб для командної роботи. У Microsoft Teams доступні як звичні, так і спеціалізовані додатки Office 365. Співробітники можуть телефонувати і проводити відеоконференції, а також працювати з документами прямо в Microsoft Teams.

Налаштування під кожную команду. Microsoft Teams пропонується можливість індивідуальної настройки функцій, які відповідають конкретним запитам і потребам. Члени команди можуть створювати окремі теми, щоб організувати спілкування. Кожна тема може бути налаштована індивідуально за допомогою функції Tabs, яка надає швидкий доступ

до часто використовуваних документів і додатків. Tabs можуть бути створені для таких сервісів Office 365, як OneNote, SharePoint і Planner, або сторонніх рішень. Також здійснена підтримка Microsoft Bot Framework, що дозволяє вбудувати в корпоративну роботу середу інтелектуальні сервіси. Попередня версія Microsoft Teams Developer Preview дає розробникам можливість почати роботу над інтеграцією з Microsoft Teams прямо зараз.

Безпека. Microsoft Teams є частиною Office 365 і пропонує безпеку і дотримання заданих політикам, реалізованих на базі Microsoft Cloud. Дані постійно проходять шифрування, багатофакторна аутентифікація забезпечує просунутий захист, щоб забезпечити безпеку даних всередині команди [8].

Огляд написаний на підставі особистого досвіду автора, а також джерел [1-8].

ЛІТЕРАТУРА

1. https://en.wikipedia.org/wiki/Microsoft_Teams - Вікіпедія Microsoft Teams
2. <https://products.office.com/en-us/microsoft-teams/group-chat-software> - Офіційний сайт Майкрософт
3. <https://zapier.com/apps/microsoft-teams> - Огляд Microsoft Teams
4. <https://support.office.com/en-us/article/welcome-to-microsoft-teams-free-6d79a648-6913-4696-9237-ed13de64ae3c?omkt=en-US&ui=en-US&rs=en-US&ad=US> - Офіційна підтримка Майкрософт стосовно пакету Office
5. <https://teams.microsoft.com/downloads> - Офіційний сайт Майкрософт
6. <https://support.office.com/en-us/article/sync-microsoft-teams-c549f1c3-eaf6-4e35-8756-464f4ae64047> - Офіційна підтримка Майкрософт стосовно пакету Office
7. <https://docs.microsoft.com/en-us/microsoftteams/platform/webhooks-and-connectors/what-are-webhooks-and-connectors> - Документація про події Майкрософт
8. <https://news.microsoft.com/ru-ru/vstrechajte-microsoft-teams/> - Зустрічайте Microsoft Teams

Шапенко Євгеній, студент бакалаврату;
Юрчук І.А., к.ф.-м.н., доцент кафедри
 кафедри Програмних систем і технологій
 Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Органайзер та поштовий клієнт MS Outlook

Microsoft Outlook — поштовий клієнт для роботи з електронною поштою, який також є повноцінним органайзером. Основні функції: календар; планувальник завдань; записник і менеджер контактів; автоматичне складання щоденника роботи з документами пакету MS Office [1-5]. Може використовуватися як окремий застосунок, так і виступати в ролі клієнта для поштового сервера Microsoft Exchange Server, що надає додаткові функції для спільної роботи користувачів однієї організації: загальні поштові скриньки, теки завдань, календарі, конференції, планування і резервування часу загальних зустрічей, узгодження документів.

Електронна пошта, календар і контакти в єдиному місці. Завдяки інтеграції продуктів Office є можливість додавати вкладення безпосередньо з OneDrive, працювати з контактами та переглядати профілі в LinkedIn.

Упорядкування справ і планування. Можливість ділитися календарями та дивитися, коли у ваших колег є вільний час, щоб планувати події й узгоджувати розклади.

Безпека. Захисту корпоративного рівня Outlook довіряють найбільші у світі організації. Він цілодобово захищає конфіденційну інформацію, не заважаючи працювати.

Аналітика. Outlook передбачає потреби людини. Рахунки, зокрема на оплату поїздки, автоматично додаються в календар, а інтелектуальні нагадування заздалегідь попереджають про те, що потрібно зробити. Функція пошуку дає змогу швидко знаходити інформацію.

Перейти до роботи в Outlook легко вже зі стартової сторінки, де знизу зліва в регульованому вікні зібрані всі функції, варто тільки вибрати необхідну.

Обравши календар є можливість розробити власний календар відсортувавши події по кольору та типу за власним бажанням.

Щодо роботи з **електронною поштою**, тут все просто. Як і інші поштовики, Outlook здатний приймати і відправляти повідомлення. При прочитанні листів ви можете бачити адресу електронної пошти відправника, час відправки і статус листи (прочитано/не прочитано). З вікна для прочитання листа ви можете за допомогою однієї кнопки перейти до написання відповіді. Також, при складанні відповіді ви можете використовувати готові шаблони листів, як вже вбудовані в програму, так і створені власноруч.

Однією з ключових особливостей поштовика від Microsoft є можливість налаштувати прев'ю листів, тобто, перші кілька рядків, які відображаються ще до відкриття листи. Дана функція дозволяє економити час, так як іноді можна відразу зрозуміти сенс листа тільки за першими кількома фразами. У більшості поштових сервісів видно тільки тема листа і перші пари слів, до того ж кількість перших видимих символів неможливо змінити.

Синхронізація та персоналізація. Вся пошта легко синхронізується з хмарними сервісами Microsoft. Тобто, якщо ви маєте аккаунт на OneDrive, то всі листи і вкладення до них ви можете переглядати з будь-якого пристрою, де навіть не інстальовано Outlook, але є Microsoft OneDrive. Ця функція може виявитися корисною, якщо ви не можете знайти потрібну вкладення в Outlook. Всі вкладення до листів зберігаються в хмарі, завдяки чому їх розмір може становити до 300 Мб. Однак, якщо ви часто прикріплюєте або отримуєте листа з великими вкладеннями, то ваше хмарне сховище може дуже швидко засмітитися ними.

Вартість. Оскільки MS Outlook входить до пакету Office 365, то безоплатне його використання можливе лише у Web-версії при реєстрації акаунту на <https://outlook.live.com>. Для повної інсталяції на комп'ютер необхідно придбати підписку Office 365. Варіанти підписок:

- 69\$ - Office 365 Personal – щорічно.
- 99\$ - Office 365 Home – щорічно, з можливістю 1 місяця пробної підписки.

Час потрібний на опанування. Outlook не потребує особливих навичок для його використання. Завдяки інтуїтивно-зрозумілому інтерфейсу та гнучкої настройки під користувача його використання є дуже зручним для будь-кого.

При проблемах з налаштуваннями можна скористатися допомогою підтримки Microsoft, яка доволі оперативно дасть відповіді на всі запитання.

Платформи та синхронізація. Для використання Outlook передбачений великий спектр різноманітних платформ, що робить його достатньо клієнтоорієнтованим.

Після виходу Outlook 98, на платформі Macintosh його замінив Microsoft Entourage, що є важливим зауваженням для потенційного користувача Apple Macintosh.

Інтеграція з іншими інструментами. В MS Outlook передбачена інтеграція з усіма інструментами базового набору Office 365, зокрема Excel та Word.

Також однією з особливостей є його інтеграція з файлообмінником Skype. Це дозволяє здійснювати відеовиклики в Skype прямо зі стартової сторінки MS Outlook.

Окрім цього в Outlook у зв'язі також йде додаток «Люди». Воно дозволяє імпортувати контактну інформацію людей з їх акаунтів в Facebook, Twitter, Skype, LinkedIn. До однієї людини можна прикріпити посилання на декілька соціальних мереж.

Рекомендована галузь використання. MS Outlook більше підійде для корпоративного використання, так як користувачам, яким не потрібно обробляти велику кількість листів і працювати з командою, дане рішення не буде достатньо корисним.

Висновки. MS Outlook це поштовий клієнт для роботи з електронною поштою, він також є повноцінним органайзером, що надає функції календаря, планувальника завдань, записника і менеджера контактів. Крім того, Outlook дозволяє відстежувати роботу з документами пакету Microsoft Office для автоматичного складання щоденника роботи.

Outlook може використовуватися як окремий застосунок, так і виступати в ролі клієнта для поштового сервера Microsoft Exchange Server, що надає додаткові функції для спільної роботи користувачів однієї організації: загальні поштові скриньки, теки завдань, календарі, конференції, планування і резервування часу загальних зустрічей, узгодження документів.

Outlook є мультиплатформенним, тому його використання буде зручним в будь-якому місці, в будь-який час.

Переваги

- Зручний і сучасний інтерфейс з якісною локалізацією;
- Спрощена робота з декількома обліковими записами;
- Можливість завантажувати великі файли як вкладення до листів;
- Є можливість купити мультиліцензію;
- Легко працювати з декількома обліковими записами одночасно.

Недоліки

- Дана програма платна;
- Можливість працювати оффлайн не до кінця опрацьована;
- Не можна робити позначки до різних email-адрес.

Рекомендується для корпоративного використання.

Матеріал підготовлений на підставі особистого досвіду автора, а також з використанням посилань на таку літературу [1-5].

ЛІТЕРАТУРА

1. <https://support.microsoft.com/ru-kz/help/2802574> - Інтеграція Microsoft Outlook 2010 та Skype
2. https://w2.outlook.com/l/upsell?WT.mc_id=PROD_OL-Web_PreAuth_May19UI - Сайт Outlook
3. <https://uk.soringpcrepair.com/outlook> - Огляд Microsoft Outlook 2016
4. <https://office.live.com/start/Outlook.aspx?ui=ru%2DRU&rs=AZ> - Outlook online
5. https://uk.wikipedia.org/wiki/Microsoft_Outlook - Стаття на Вікіпедії про Outlook

Рибалочка Михайло, студент бакалаврату;
Супрун О.М., к.ф.-м.н., доцент кафедри
 кафедри Програмних систем і технологій
 Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Confluence – програмне забезпечення командної роботи

Confluence - це зручний і просунутий wiki движок від компанії Atlassian [1-3]. Він дозволяє організувати внутрішній інтернет портал і дати доступ до нього всім користувачам - для редагування або для читання.

Confluence - це програма для співпраці, розроблена та опублікована австралійською програмною компанією Atlassian. Atlassian написав Confluence мовою програмування Java і вперше опублікував її в 2004 році. Confluence Standalone поставляється із вбудованим веб-сервером Tomcat та базою даних hsql, а також підтримує інші бази даних.

Компанія продає Confluence як корпоративне програмне забезпечення, ліцензоване як локальне програмне забезпечення, так і програмне забезпечення як послуга, що працює на AWS.

Історія. Atlassian випустив Confluence 1.0 25 березня 2004 року, заявивши, що його метою було побудувати "додаток, побудований відповідно до вимог системи управління знаннями підприємства, не втрачаючи при цьому суттєвої, потужної простоти вікі".

У 2014 році Atlassian випустив Центр обробки даних Confluence, щоб додати високу доступність з балансуванням навантаження через вузли в кластерній установці.

Книга Social Marketing Marketing for Dummies у 2007 році вважала Confluence "новим соціальним програмним забезпеченням для підприємств", яке "стає усталеним гравцем", простий у налаштуванні та використанні "та" виняток із правила", що можливості пошуку програмного забезпечення вікі не працюють добре.

eWeek цитував у 2011 році такі нові функції у версії 4, як автоматичне форматування та автоматичне доповнення, уніфіковані вікі та WYSIWYG, сповіщення в соціальній мережі та інтеграція мультимедійних файлів та перетягування. Випадки використання включають базове спілкування на підприємстві, робочі місця для співпраці для обміну знаннями, соціальних мереж, управління персональною інформацією та управління проектами. Німецька газета Computerwoche від IDG Business Media порівнює його з Microsoft SharePoint і вважає це «хорошою відправною точкою» як платформи для співпраці в соціальному бізнесі, тоді як SharePoint краще підходить для компаній з більш структурованими процесами.

Confluence включає налаштування шаблонів CSS для стилів та форматування для всіх сторінок, включаючи ті, що імпортуються з документів Word. Вбудований пошук дозволяє запити за датою, автором сторінки та типом вмісту, наприклад графікою.

Інструмент має додатки для інтеграції зі стандартними форматами, з гнучким програмованим API, що дозволяє розширювати. Програмне забезпечення є релевантним інструментом для окреслених вимог, які можуть бути пов'язані із завданнями в трекері проблем Jira тієї ж компанії.

Припинення розмітки вікі. Станом на версію 4.0, у 2011 році Confluence припинила підтримку мови розмітки вікі. Це призвело до бурхливої дискусії користувачів попередніх версій, які заперечили проти змін. У відповідь Atlassian надав редактор вихідного коду як плагін, що дозволяє досвідченим користувачам редагувати базове джерело документів на основі XHTML. Однак, хоча нова розмітка джерела заснована на XHTML, вона не сумісна з XHTML.

Крім того, розмітка вікі може бути введена в редактор, а функція автозаповнення та автоматичного форматування Confluence перетворює розмітку вікі у новий формат. Після конверсії в режимі реального часу вміст не можна редагувати як вікі-розмітку знову.

Безпека. Дані хмарного злиття шифруються під час транзиту та в спокої.

Обмеження

- Дві сторінки не можуть мати однакову назву в одному просторі імен, навіть якщо вони мають різних батьків. Це може виявитися незручним, якщо злиття призначене для документації, яка має повторювані заголовки.
- Не існує стандартного способу додавання підписів до зображень, є лише обхідні шляхи.

Налаштування продажів. Atlassian не має традиційної команди з продажу, натомість покладається на свій веб-сайт.

Вартість

- Для ознайомлення (персонально) - безкоштовно
- Standard - 5 доларів/місяць
- Premium - 10 доларів/місяць
- Також існують корпоративні версії ціну яких обговорюють окремо згідно потреб клієнта.

Час потрібний на опанування. Confluence має дуже прости інтуїтивний інтерфейс, на опанування якого потрібно декілька хвилин. Функціональність додатку строго визначена та не потребує особливих навичок в опануванні.

Платформи та синхронізація. Існують версії для Windows, Linux, macOS, iOS, Android.

Інтеграція з іншими інструментами. В останніх версіях Confluence перетворилася на частину інтегрованої платформи співпраці і була адаптована для роботи спільно з Джирою та іншими програмними продуктами Atlassian, включаючи Bamboo, Clover, Crowd, Crucible та FishEye.

Рекомендована галузь використання. Бухгалтерія, фінанси, розробка ПЗ, документообіг, цей список можна продовжувати до бескінечності.

Висновок. Confluence це сучасна система для документообігу з можливостями систем контролю версій та обговорення. Система доступна всім бажаючим для ознайомлення та широко використовується в ІТ та державних проектах.

Текст написаний на підставі особистого досвіду автора, а також з використанням посилань на наступну літературу [1-3].

ЛІТЕРАТУРА

1. <https://www.atlassian.com/ru/software/confluence> - Офіційна сторінка Confluence
2. [https://en.wikipedia.org/wiki/Confluence_\(software\)](https://en.wikipedia.org/wiki/Confluence_(software)) - Стаття на вікіпедії про Confluence
3. <https://confluence.atlassian.com/alldoc/atlassian-documentation-32243719.html> - офіційна документація про Confluence

Шестаков Андрій, студент бакалаврату;
Поляков С.А., к.ф.-м.н., доцент кафедри
 кафедра Програмних систем і технологій
 Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Програма телекомунікацій Skype

Skype – програма телекомунікацій, яка спеціалізується на наданні відеочату та голосових дзвінків між комп'ютерами, планшетами, мобільними пристроями, консоллю Xbox One та смартгодинниками через Інтернет [1, 2]. Skype також надає послуги обміну миттєвими повідомленнями. Користувачі можуть передавати текст, відео, аудіо та зображення. Skype дозволяє здійснювати відеоконференції.

Програма дозволяє легко провести конференц-дзвінки (до 25 голосових абонентів, включаючи ініціатора), відеодзвінки (у тому числі відеоконференції до 10 абонентів), а також забезпечує передачу текстових повідомлень (чат) і передачу файлів. Є можливість створити зображення з веб-камерами, передавати зображення з екрану монітора, а також створювати та відправляти відеозаписи користувачів настільних версій програм.

Доступність. Skype намагались заблокувати в Азербайджані, Білорусі, Німеччині, Китаї, Росії та Ефіопії. Наразі влада деяких країн і досі намагається блокувати користувачів цієї програми, проте вона все ще актуальна й поширена по всьому світові.

Масштабованість за кількістю користувачів. Skype підтримує до 150 користувачів у одному дзвінку для звичайного плану. Наразі Максимальна кількість одночасних викликів або каналів викликів обмежена до 300 на SIP-профіль. Однак можна створити стільки профілів SIP, скільки вам подобається. За допомогою Skype Connect ви також можете скористатися "розривами". Кількість вхідних викликів, які може підтримувати ваш бізнес, залежить від того, як ви налаштували Skype Connect, інфраструктуру мережі та підключення до Інтернету.

Ступінь шмарності. У таблиці 1 наведено вимоги до мінімальної швидкості завантаження й передавання, а також рекомендовані швидкості, що забезпечують оптимальну продуктивність.

Таблиця 1.

Швидкості завантаження й передавання

Тип виклику	Мінімальна швидкість завантаження й передавання	Рекомендована швидкість завантаження й передавання
Виклики	30 кбіт/с; 30 кбіт/с	100 кбіт/с; 100 кбіт/с
Відеовиклики/спільний перегляд екрана	128 кбіт/с; 128 кбіт/с	300 кбіт/с; 300 кбіт/с
Відеовиклики(висока якість)	400 кбіт/с; 400 кбіт/с	500 кбіт/с; 500 кбіт/с
Відеовиклики(HD)	1,2 Мбіт/с; 1,2 Мбіт/с	1,5 Мбіт/с; 1,5 Мбіт/с
Відеоконференції(3 учасники)	512 кбіт/с; 128 кбіт/с	2 Мбіт/с; 512 кбіт/с
Відеоконференції(5 учасників)	2 Мбіт/с; 128 кбіт/с	4 Мбіт/с; 512 кбіт/с
Відеоконференції(понад 7 учасників)	4 Мбіт/с; 128 кбіт/с	8 Мбіт/с; 512 кбіт/с

Якщо ви ввійшли в Skype, але не здійснюєте жодних викликів, Skype використовуватиме приблизно 0–4 кбіт/с. Якщо ви здійснюєте виклик, Skype використовує приблизно 24–128 кбіт/с.

Вартість. Загалом Skype безкоштовний, проте для дзвінків на мобільні та стаціонарні номери є спеціальні тарифи. Наприклад, вартість безлімітних дзвінків по Північній Америці на місяць буде коштувати майже 8,5\$ – так само, як і до Росії. А ось з

абонентами з усього світу можна зв'язатися за майже 17\$, а ось якщо телефонуєте без підключення до тарифу, то можете заплатити вже від 5 до 25\$.

Таблиця 2.

Вимоги до апаратних засобів

Windows для настільних ПК	
Версія	Windows 10 версії 1507 або новішої Windows 8.1 Windows 8 Windows 7 (підтримуються 32- і 64-розрядна версії)
Процесор	Принаймні 1 ГГц
Оперативна пам'ять (ОЗП)	Принаймні 512 МБ
Додаткове програмне забезпечення	DirectX 9.0 або новішої версії
Mac	
Версія	Mac OS X 10.10 або новіша версія
Процесор	Процесор Intel принаймні 1 ГГц (Core 2 Duo)
Оперативна пам'ять	Принаймні 1 ГБ
Додаткове програмне забезпечення	Найновіша версія QuickTime

Час потрібний на опанування. Skype надає можливість швидко опанувати програмний продукт. Для повного розуміння роботи з програмою потрібно лише декілька хвилин. Програма Skype підтримує технології спеціальних можливостей, зокрема невізуальні екрани, доступні для Windows і MacOS.

Платформи та синхронізація. Існують версії Skype для Windows, Linux, macOS, iOS, Android, Windows Mobile, Windows Phone, PSP, Symbian OS, Java (мобільних телефонів).

Рекомендована галузь використанняБезкоштовні послуги

- SkypeCast (від англ. Skype - програма VoIP і broadcasting - широкомовлення, іноді використовується скорочене «каст») - вид голосового спілкування між групою користувачів програми Skype (до 150 осіб). Зовні схожий з конференц-дзвінком, однак, на відміну від нього, встановлюється через центральний сервер, внаслідок чого не пред'являє високих вимог до пропускної здатності каналу користувача, який ініціював розмову. Послуга більше не надається.
- Skype Voicemail - голосова пошта. Дозволяє записувати вхідні повідомлення, коли користувач не в мережі, і працює як автовідповідач.
- Skype Me. Зв'язавши користувачів по всьому світу за допомогою голосу, Skype дав можливість людям з різних країн спілкуватися один з одним. Для полегшення цього завдання Skype пропонує встановлювати статус SkypeMe, який вказує, що даний абонент відкритий для дзвінків з усього світу. Зараз видалена

Платні послуги

- SkypeOut - дзвінки на телефони. Дозволяє здійснювати вихідні дзвінки на стаціонарні і мобільні телефони в більшості країн світу. Оплата похвилинна, диференційована. Дзвінки на безкоштовні номери (такі, як +1 800 в США) безкоштовні, причому скористатися ними можуть навіть користувачі, які не оплачували послугу SkypeOut.
- SkypeIn - онлайн-номер. Дозволяє отримувати телефонні дзвінки від користувачів традиційних телефонних мереж.
- Номер Skype To Go - спеціальний номер доступу, на який можна зателефонувати з будь-якого телефону (стаціонарного або мобільного) для того, щоб зв'язатися з іншим номером за встановленими тарифами, грошові кошти при цьому знімаються з рахунку Skype.
- Відправлення SMS - можливість відправляти повідомлення з програми Skype на мобільні телефони.

Послуги для бізнесу

Для корпоративного сегменту ринку Skype пропонує послугу Skype для SIP Open Beta і її більш функціональну різновид Skype Connect. Необхідна кількість ліній оплачується щомісяця.

Дані послуги дозволяють компаніям забезпечити прямий доступ зі своєї корпоративної мережі зв'язку прямо в мережу Skype, а саме: приймати на офісні телефони дзвінки з мережі Skype і здійснювати з офісних телефонів дзвінки в мережу Skype. Доступ до мережі Skype реалізується за допомогою VoIP-шлюзів (через офісні аналогові або цифрові АТС) або через IP-АТС компанії. Сумісне зі Skype обладнання сьогодні випускається такими постачальниками, як Avaya, Cisco, AddPac.

ЛІТЕРАТУРА

1. <https://support.skype.com/en/skype/all> - Офіційна підтримка Skype (питання/відповідь)
2. <https://en.wikipedia.org/wiki/Skype> - Стаття про Skype на Вікіпедії

Балацкий Дмитро, студент бакалаврату;
Меркулова К.В., к.т.н., доцент кафедри
 кафедри Програмних систем і технологій
 Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Dropbox – програмне забезпечення командної розробки

Dropbox - файловий хостинг компанії Dropbox Inc., що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт. Дозволяє користувачеві розмішувати файли на віддалених серверах за допомогою клієнта або з використанням веб-інтерфейсу через браузер. При установці клієнтського програмного забезпечення Dropbox на комп'ютері створюється і синхронізуються папка. Для спільної роботи над проектами сервіс має можливість створення «Shared» папок для загального доступу осіб, які мають різні облікові записи на сервісі. Доступна автоматична синхронізація файлів і папок і зберігання версій з можливістю відкату

Безпечний спільний доступ до папок усіх учасників робочої групи

Безпека спільного доступу до папок. З Dropbox можна надавати доступ до папок і запрошувати в них не тільки колег, але і тих, хто не працює в компанії.

Збереження папок в одному місці. Dropbox використовує єдине захищене сховище для всіх ваших папок і файлів.

Контроль файлів в режимі реального часу. Вам будуть приходити повідомлення, коли хтось переглядає, редагує, переміщує або видаляє файл або папку.

Безпечний обмін посиланнями

Обмін файлами за допомогою безпечних посилань.

Легкість обміну посиланнями в Dropbox. Мобільний додаток Dropbox дозволяє ділитися посиланнями на одні і ті ж файли, поки ви знаходитесь в дорозі, безпосередньо зі свого пристрою iPhone або Android.

Обмін посиланнями з додатковим захистом.

- Будь-який користувач Dropbox може в будь-який час відключити можливість доступу, видаливши загальну посилання на свій файл
- Користувачі Dropbox Professional і Business можуть додати пароль, встановити термін дії і відключити можливість скачування для будь-якої загальної посилання
- Адміністратори Dropbox Business можуть встановити обмеження на доступ до загальних посиланнях тільки для певних груп або учасників робочої групи

Віддалена робота і офлайн-синхронізація

Робота при відсутності Wi-Fi.

Доступ до файлів з усіх пристроїв. Програма Dropbox забезпечує можливість автоматичної синхронізації файлів на сервері з файлами на ПК з операційною системою Windows або Mac для роботи з ними в автономному режимі.

Доступність папок. Будь-яку папку можна цілком синхронізувати з папкою на комп'ютері або мобільному пристрої для роботи в автономному режимі.

Нотатки до фото і PDF

Коментарі прямо в файлах. Dropbox дозволяє коментувати будь-яке зображення або текст в режимі попереднього перегляду, включаючи файли Excel, Photoshop і Sketch, навіть якщо у вас немає відповідного програмного забезпечення.

Анотація PDF в програмі Acrobat з будь-якого пристрою. Завдяки інтеграції з Adobe Acrobat, залишати різні типи приміток на документах PDF стало простіше. Додаткові інструменти для роботи з файлами PDF більше не потрібні.

Обмін відгуками в режимі реального часу без довгих ланцюжків електронних повідомлень з уточнюючими питаннями по проектам.

Управління завданнями для створення простих списків справ

Відмітка виконаних завдань в своєму списку під час роботи.

Контроль виконавців. Використовуйте різні інструменти управління завданнями прямо з Paper, наприклад розподіл завдань або постановку термінів виконання собі і учасникам робочої групи. У міру наближення термінів задачі Paper розішле автоматичні

нагадування всім учасникам, які допоможуть їм бути в курсі подій і відстежувати хід виконання.

Підтримка мозкової атаки

Швидка фіксація ідей. На відміну від інших шаблонів для ідей, цей документ Dropbox Paper не вимагає попереднього форматування. Він створений таким чином, щоб записати хід думки, поки вас не покинуло натхнення.

Запрошення колег по команді. Даний хмарний шаблон концепт-плану або інтелекту карти допомагає в розробці ідей при самостійному або колективному мозковому штурмі.

Сумісна творчість в режимі реального часу. Включені в створену вами інтелекту карту учасники зможуть спостерігати за вашими ідеями в міру того, як ви їх записуєте, можуть залишати відгуки в коментарях і пропонувати нові ідеї.

Перегляд файлів в режимі реального часу

Відкривайте і переглядайте файли без додаткового програмного забезпечення. При використанні Dropbox для перегляду файлів можна переглядати і коментувати файли PDF, а також переглядати, коментувати і надавати загальний доступ до файлів самих різних типів, включаючи файли з розширенням .csv і .xlsx, таблиці Excel, презентації PowerPoint, а також файли Photoshop і Sketch. При використанні Dropbox для перегляду файлів можна переглядати найрізноманітніші відео-, аудіо- та текстові файли з будь-яких пристроїв.

Додавайте коментарі до файлів з будь-яких пристроїв.

Можливість спільної роботи з будь-якого місця. Ви завжди будете в курсі всіх важливих подій незалежно від того, де ви знаходитесь. Функція перегляду файлів Dropbox дозволяє автоматично отримувати повідомлення про всі нові коментарі до ваших файлів, навіть якщо ви не відзначені в коментарях.

Вартість

Standard: 12,50 \$ / користей. в місяць, починаючи з 3 користувачів

- 5 ТБ місця для безпечного зберігання файлів
- Прості інструменти для надання доступу та спільної роботи
- Відновлення файлів протягом 180 днів
- Єдиний вхід для адміністратора, щоб управляти декількома робочими групами
- Консоль адміністрування і журнал аудиту
- 256-бітове шифрування AES і шифрування SSL / TLS
- настроюються дозволи для доступу
- Підгрупи, керовані компанією і користувачами
- Віддалене стирання даних
- Двохетапна перевірка справжності (2FA)
- Розумна синхронізація
- Інструменти адміністрування Dropbox Paper
- Інтеграція з Microsoft Office 365
- Безлімітний доступ через API до партнерських платформ для підвищення безпеки та ефективності

- 1 мільйон викликів API в місяць партнерам по передачі даних
- Прискорена технічна підтримка в чаті, по ел. поштою і телефону (в робочий час)

Advanced: 20 \$ / користей. в місяць, починаючи з 3 користувачів

- Стільки місця, скільки потрібно вашим співробітникам
- Вдосконалені функції управління і безпеки
- Все, що є в Standard
- Потужні засоби адміністрування
- Адміністратори різних типів
- Розширені засоби управління доменом
- Інтеграція функції єдиного входу (SSO)
- Управління запрошеннями
- Підтвердження доменів
- Підтвердження пристроїв
- Відстеження подій, пов'язаних з файлами

Enterprise: Ціна встановлюється за особистою домовленістю

- рішення налаштовуються персонально
- Персоналізована технічна підтримка, щоб допомогти адміністраторам керувати аккаунтом всієї організації
- Все, що є в Advanced
- Централізована консоль адміністрування, що дозволяє одночасно переглядати всі ваші робочі групи Dropbox Business і управляти ними
- Акаунти в домені
- Мережевий контроль
- Управління віддаленим доступом на корпоративному рівні (EMM)
- Статистика домену
- Покращена підготовка кінцевих користувачів і адміністраторів
- Цілодобова підтримка по телефону

Час потрібний на опанування. Час опанування может зайняти кілька днів в залежності вашого інтересу або потреби в опануванні. Не важливо, новий ви користувач або просто хочете освіжити свої знання про Dropbox, інформацію ви можете отримати в нашому курсі для самостійного навчання або віртуальних уроках з інструктором.

Платформи та синхронізація. Існують версії для Windows, Linux, macOS, iOS, Android, Windows Mobile, Windows Phone, PSP, Symbian OS, Java (мобільних телефонів).

Інтеграція з іншими інструментами. Autodesk, Jira, Slack, Okta, Canva, Final Draft, Frame.io, Marvel, Getty Images, Shift, Widen

Висновки

Плюси dropbox. Можливість спільної роботи з будь-якого місця. Ви завжди будете в курсі всіх важливих подій незалежно від того, де ви знаходитесь. Функція перегляду файлів Dropbox дозволяє автоматично отримувати повідомлення про всі нові коментарі до ваших файлів, навіть якщо ви не відзначені в коментарях. Автоматичні повідомлення дозволяють вам бути в курсі всіх дій ваших колег.

- Автоматичне визначення швидкості інтернет-каналу
- Історія зміни файлу
- Відновлення видалених файлів
- Можливість поділитися файлами і папками
- Є помісячна оплата
- У зв'язку з високою популярністю є інтеграція з багатьма додатками і пристроями
- Є додаток для Linux

Мінуси dropbox

- Не можна синхронізувати папки за межами каталогу Dropbox
- Немає додатку для Windows Phone
- Синхронізація для Android зроблена незручно. Але можна скористатися будь-яким файловим менеджером на телефоні

Текст написаний на підставі особистого досвіду автора, а також з використанням посилань на наступну літературу [1, 2].

ЛІТЕРАТУРА

1. <https://www.dropbox.com/> - Офіційний сайт Dropbox
2. <https://ru.wikipedia.org/wiki/Dropbox> - Стаття про [Dropbox](https://www.dropbox.com/) на Вікіпедії

Дяченко Ігор, студент бакалаврату;
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Evernote - інструмент для командної розробки програмного забезпечення

Evernote — онлайн-сервіс та ряд додатків для різних платформ, призначених для збереження, синхронізації та пошуку приміток, в тому числі текстових записів, веб-сторінок, списку задач, фотографій, малюнків, електронної пошти та ін. [1, 2, 3]. Власник продукту – компанія Evernote Corporation. Компанія була створена в 2000 році, вихідцем з Казахстану – Степаном Пачіковим.

Призначення Evernote залежить від стилю використання: Особистий(сімейний) і Корпоративний. Особистий Evernote виконує функцію ефективного записника, який дозволяє дуже швидко зберігати інформацію(завдяки веб-технологіям) і фільтрувати(система тегів). Корпоративний Evernote додає функцію документообігу до функціоналу особистого Evernote.

Девіз Evernote – «Пам'ятати все».

Загалом Evernote - це система створення приміток, які мають багатий функціонал: створення сповіщень, фільтрація по темі, sharing(соц. медіа і так далі), хмарне зберігання, історія змін і так далі. При створенні примітки є можливість повісити оповіщення, яке буде працювати залежно від платформи.

Getting Things Done (GTD) – це методика підвищення особистої ефективності, створена консультантом Девідом Аланом, і описана в його однойменній книзі. Метод базується на ідеї, що для ефективнішого планування ніж природне людині необхідно залучати додаткові засоби, а саме зовнішні носії інформації і додаткові методики систематизації. У природний спосіб люди використовують для планування префронтальну кору мозку. Розум звільняється від роботи запам'ятовування всього, що необхідно зробити, і може концентруватися на власне виконанні цих завдань.

Для використання цієї методики, найчастіше використовується саме Evernote, як інструмент планування, систематизації і запам'ятовування. Більшість сучасних бізнес-школ рекомендують саме Evernote, через його зручність і доступність.

Доступність: На момент 2019 року не було задокументовано ні одного випадку про блокування або серйозні технічні проблеми сервісу Evernote.

Ступінь хмарності: Для роботи Evernote використовує локальну базу даних, але через певний період часу відправляє інформацію на сервер. Також доступна функція ручної синхронізації і вимкнення автоматичної синхронізації.

Не виключено, що при непередбачуваній ситуації можлива втрата певних останніх змін, так як Evernote кожні 15 хвилин синхронізується з хмарним сховищем автоматично.

Масштабованість за кількістю користувачів: Як було зазначено раніше в пунктах «Основна функціональність ...», Evernote використовується в двох режимах: Особистий і Корпоративний. В особистому режимі зрозуміло, що користувач один, але може мати багато девайсів. В корпоративному режимі користувачів багато і девайсів багато. Як було зазначено раніше, корпоративний режим має всі функції для контролю користувачів і доступів. Також не потрібно забувати про синхронізацію з CRM системами, що дозволяє відслідковувати і створювати потрібні примітки на основі бізнес інформації.

Теги і Блокноти в Evernote: В Evernote система розподілення і фільтрування приміток працює на основі тегування: при створенні примітки є можливість вказати мітки, які віднесуть примітку до певної групи. Групи тегів створюються в додатку і в інтерактивному режимі можуть перебудовуватися користувачем. Ця система набагато зручніше ніж звична нам файлова система, бо дозволяє зразу прив'язувати примітку до багатьох груп не думаючи, як ці групи повинні структуруватися.

Блокноти в Evernote існують, щоб розподіляти примітки не тільки по групам, а й по семантичному значенні; тобто це як раз і вносить дуже добре нам знайому файлову систему, але лише на верхньому рівні, що дозволяє отримати переваги і позбутися мінусів.

Висновок: Evernote використовує поєднання двох систем структурування інформації для досягнення зручності, швидкості і гнучкості.

Час потрібний на опанування. Evernote має інтуїтивно зрозумілий інтерфейс, тому розібратися з ним можна за 20 хвилин. Також Evernote має чудовий сайт з посібниками (<https://help.evernote.com/>) і графічними прикладами. Найкращий варіант - це скачати і запустити, тому що при запуску, Evernote вмикає програму інтерактивного навчання і все показує на практиці.

Платформи та синхронізація. Evernote доступний: Windows, IOS, macOS, Android. На жаль, сервіс не доступний на операційній системі Linux. Для таких випадків доступна Web версія додатку, але з обрізаними можливостями.

Інтеграція з іншими інструментами

- Підключення Google Drive, Outlook і Slack
- Підключення до CRM систем (Salesforce)

Вартість та основна функціональність різних конфігурацій

Основна функціональність плану «Evernote basic»:

- Створення і форматування таблиць в примітках.
- Зберігання копії веб-сторінок.
- Аудіо-примітки
- Прикріплення файлів
- Пошук
- Сортування
- Дозвіл на перегляд і редагування примітки ззовні.

«Evernote basic» - безкоштовно. Розмір сховища – 65MB на місяць. Максимальний розмір примітки - 25MB. Кількість девайсів – 2. Наявність проектів – відсутня.

Висновок: План підходить для створення маленьких приміток для особистого використання.

Основна функціональність плану «Evernote Premium»:

«Evernote basic» +

- Підтримка електронних листів
- Сканування візиток і створення приміток
- Редагування і робота з PDF файлами
- Пошук в прикріплених документах
- Підтримка відео-приміток
- Створення презентацій на основі приміток
- Історія змін примітки
- Підключення Google Drive, Outlook і Slack

«Evernote Premium» - 75 грн. Розмір сховища – 10GB на місяць. Максимальний розмір примітки - 200MB. Кількість девайсів – не обмежено. Наявність проектів – відсутня.

При використанні «Evernote Premium» стає доступним чат підтримки.

Висновок: План підходить для активного використання Evernote, як інструмента для приміток, планування, зберігання і контролю важливої інформації.

Основна функціональність плану «Evernote BUSINESS»:

«Evernote Premium» +

- Підтримка розподілених блокнотів і приміток
- Рекомендації приміток
- Підключення до CRM систем (Salesforce)
- Проекти для сумісної співпраці
- Контроль доступу
- Дошка новин
- Сумісне редагування в режимі реального часу
- Централізоване керування аккаунтами
- Історія активності аккаунта
- Юридичні права на вміст аккаунта
- Єдиний вхід

«Evernote Business» - 300 грн. Розмір сховища – 10GB на одного користувача на місяць. Максимальний розмір примітки - 200MB. Кількість девайсів – не обмежено. Наявність проектів – доступна.

Цей план включає в себе функціонал розподіленої роботи для великої кількості працівників. Головна функція – це керування доступами і безпекою, що в контексті компанії займає майже найважливіше значення.

Висновок: План підходить для активного використання Evernote, як інструмента для приміток, планування, зберігання і контролю важливої інформації.

Загальний висновок

Evernote має інтуїтивно зрозумілий інтерейфс, тому розібратися з ним можна за 20 хвилин. Також Evernote має чудовий сайт з посібниками (<https://help.evernote.com/>) і графічними прикладами. Найкращий варіант - це скачати і запустити, тому що при запуску, Evernote вмикає програму інтерактивного навчання і все показує на практиці.

Текст написаний на підставі особистого досвіду автора, а також з використанням посилань на наступну літературу [1, 2, 3].

ЛІТЕРАТУРА

1. <https://evernote.com> - Офіційний сайт Evernote
2. <https://help.evernote.com> - Офіційна підтримка Evernote (питання/відповідь)
3. <https://en.wikipedia.org/wiki/Evernote> - Стаття про Evernote на Вікіпедії

Ткаченко М.В., к.т.н., асистент кафедри
Федоренко Р.М., к.е.н., асистент кафедри
Шевченко В.Л., д.т.н., проф., професор кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

AcceleratedSAP (ASAP)

Методологія впровадження AcceleratedSAP (ASAP) – це інструмент, представлений компанією SAP в 1996 році для швидкого впровадження системи. Методологія ASAP – це структурний підхід до впровадження, який значно прискорює просування проекту і забезпечує ефективне навчання користувачів, вичерпну документацію і ясно складені мережеві графіки на всіх стадіях проекту. В основу методології ASAP закладено жорстку каскадну модель – кожна фаза включає формальну процедуру запуску та формальну фазу завершення, що включає набір контрольних заходів. Методологія не передбачає переміщення між фазами в зворотному напрямку, впровадження здійснюється "за один прохід".

З огляду на універсальність і, як наслідок, складність продуктів компанії SAP, була розроблена методологія впровадження продуктів SAP – Accelerated SAP, на основі якої була розроблена методологія ValueSAP [1]. В рамках ValueSAP клієнтам пропонується ряд заходів, пов'язаних з інструментами системи SAP – карти рішень SAP, карти технологій SAP, карти послуг та методологія ASAP для підвищення ефективності впровадження SAP в компанії. Однак методологія ASAP постійно удосконалюється і на цей час (2019 рік) існує 8 версія Accelerated SAP [2]. Ця методологія охоплює всі етапи впровадження системи SAP – в тому числі оцінку, впровадження та подальші удосконалення системи.

Дорожня карта ASAP (Roadmap) була розроблена для підтримки зусиль проектних команд, спрямованих на планування, управління та завершення проектів на всьому шляху впровадження та експлуатації рішень SAP. ASAP Roadmap складається з п'яти фаз:

1. Project Preparation (Підготовка проекту) – проект формально ініційований і планування йде повним ходом.
2. Business Blueprint (Концептуальне проектування) – проектна команда визначає вимоги та здійснює концептуальне проектування рішень.
3. Realization (Реалізація) – рішення побудовано та його інтеграція протестована; тестування продуктивності планується.
4. Final Preparation (Фінальна підготовка) – кінцеві користувачі навчені; це завершальна перевірка перед переходом на нове системне рішення.
5. Go Live and Support (Експлуатація та підтримка) – рішення отримало схвалення, почалася безперервна підтримка. Проект завершено.

Кожна фаза візуалізується в розрізі галузей знань (knowledge areas) проекту впровадження. В рамках кожної окремої галузі знань виконуються групи робіт (deliverable group), що складаються з окремих робіт (deliverable), які надають на виході (output) результат, що найчастіше являє собою певний робочий документ або документи – прискорювачі (accelerators) в термінології ASAP. Саме орієнтованість методології ASAP на створення великої кількості документації (accelerators) і надала назву «AcceleratedSAP».

На першій фазі – Підготовка проекту (Project Preparation) – здійснюється початкове планування та загальна підготовка до проекту (табл. 1). Визначаються межі проекту, формулюються цілі, позначаються пріоритети.

Наступна фаза – Концептуальне проектування (Business Blueprint) – проводиться створення концептуального проекту IC, який описує всі вимоги та модель системи (табл. 2).

На третій фазі – Реалізація (Realization) – реалізується створений раніше концептуальний проект (табл. 3). Всі необхідні зміни впроваджуються в систему. На завершення фази проводиться тестування та формування релізної версії.

Наступна фаза – Фінальна підготовка (Final Preparation) – завершується підготовка ІС до промислової експлуатації, навчання персоналу та роботи з тестування, мають бути вирішені все критично важливі проблеми та недоліки системи (табл. 4).

Таблиця 1

Цілі, етапи та результати фази «Підготовка проекту»

Цілі	Етапи	Результати
Визначити цілі проекту з урахуванням загальної стратегії підприємства. Розглянути і зрозуміти галузь впровадження.	Ініціація. Розроблення стратегії і стандартів.	Завершено підготовку проекту. Визначено область дії проекту.
Сформувати план і порядок впровадження. Підготувати стандарти і процедури. Провести організацію проекту. Визначити необхідне ресурсне забезпечення.	Планування ІТ-інфраструктури. Планування навчання користувачів.	

Таблиця 2

Цілі, етапи та результати фази «Концептуальне проектування»

Цілі	Етапи	Результати
Створити концептуальний проект та бізнес-модель ІС. Визначити формат управління змінами. Переглянути стартові цілі та завдання проекту. Виділити предметну область проекту. Переглянути розклад проекту та порядок впровадження. Переглянути технічний проект ІС.	Управління проектом. Навчання персоналу та створення документів. Первинне визначення вимог. Управління змінами в організації. Деталізація вимог. Розширення функціональності та розробка змін. Визначення вимог щодо безпеки. Проектування ІТ-інфраструктури. Встановлення ІС для розробки.	Концептуальний проект ІС розроблений. Бізнес-модель ІС розроблена. Конфігурація, модифікації та функціональність ІС визначені. ІС встановлена для розробки.

Таблиця 3

Цілі, етапи та результати фази «Реалізація»

Цілі	Етапи	Результати
Провести конфігурування ІС. Розширити функціональність ІС. Протестувати ІС. Визначити стратегію здавання ІС. Сформувати та встановити ролі і рівні доступу.	Управління проектом. Управління змінами. Навчання персоналу і створення документів. Розробка основних елементів. Конфігурація основних елементів. Інсталяція ІС для проведення тестів. Планування здавання ІС. Впровадження засобів безпеки. Підсумкова конфігурація. Тестування інтеграції. Інсталяція виробничої версії. Тестування навантаження.	ІС реалізована. ІС протестована. Виробнича версія ІС успішно встановлена.

Таблиця 4

Цілі, етапи та результати фази «Фінальна підготовка»

Цілі	Етапи	Результати
Закінчити роботи над ІС. Встановити системи та процедури підтримки кінцевих користувачів ІС.	Управління проектом. Завершення підготовки замовника до прийому ІС. Завершення підготовки ІС. Встановлення системи підтримки користувачів. Здавання ІС.	ІС готова до промислової експлуатації.

Завершальна фаза – Експлуатація та підтримка (Go Live and Support) – проводиться остаточний перехід до промислового застосування ІС, удосконалюються підтримка та основні операційні види діяльності (табл. 5). Фаза включає два окремих періоди: завершення проекту та удосконалення. У період завершення проекту

вирішуються всі нові проблеми паралельно з введенням команди технічної підтримки в курс справи. Відбувається формальне закриття проекту і передача формалізованих знань. У період удосконалення вирішуються нові проблеми в ІС та проводиться її моніторинг.

Таблиця 5

Цілі, етапи та результати фази «Експлуатація та підтримка»

Цілі	Етапи	Результати
Провести формальне закриття проекту. Забезпечити роботу системи підтримки. Завершити проект фактично.	Управління проектом. Підтримка ІС на ранніх стадіях. Закриття проекту.	ІС впроваджена в промислову експлуатацію та використовується. Система підтримки ефективно налагоджена.

Маршрутна карта ASAP інтегрована у SAP Solution Manager, що постачається безкоштовно з будь-яким набутим комерційним продуктом SAP.

ASAP Implementation Roadmap доступна клієнтам і партнерам на порталі підтримки SAP Support Portal у вигляді веб-додатку.

Інструментарій ASAP:

- SAP Solution Manager – це основний інструмент, який свого часу був розроблений для підтримки команди проекту на всьому протязі впровадження рішень SAP доки рішення не стане продуктивним (буде введено в експлуатацію). SAP Solution Manager пропонує централізований доступ до інструментів, методології та попередньо налаштованому вмісту, які можна використовувати протягом усього процесу впровадження рішень SAP.
- SAP Solution Composer (компонувальник рішень) – синхронізує бізнес потреби клієнта з рішеннями компанії SAP.
- Маршрутні карти SAP – пропонуються маршрутні карти ASAP, складені з урахуванням специфіки конкретного проекту.

Функціональність

ведення структури інвестиційної програми;
підтримка фінансового планування і контролю;
можливість обліку натуральних показників проекту;
підтримка пріоритезації завдань;
класифікація завдань;
автоматизація процесу відбору інвестиційних проектів за визначеними факторами ранжування;
складення та підтримка моделі життєвого циклу проекту, у тому числі планування точок прийняття рішення;
повна підтримка календарного планування, у тому числі за різними строками (прогнозними, ранніми, затвердженими, фактичними тощо);
управління проектними ресурсами, призначення співробітників та делегування їм повноважень за проектом;
облік трудовитрат;
робота з проектними документами – формування папок документів, реєстрація їх статусу, зберігання та спільна робота з документами;
автоматизація процесу управління змінами;
зберігання історії змін тощо.

Вартість. Сама по собі технологія безкоштовна. Основні витрати пов'язані з закупівлею ядра, функціональності системи та ліцензій на програмне забезпечення.

Час потрібний на опанування. З огляду на стислі терміни проектів по впровадженню систем SAP, компанія SAP виділяє навчання як вкрай важливий компонент успіху будь-якого проекту. Компанія SAP пропонує широкий спектр курсів навчання, які охоплюють всі організаційні аспекти будь-якого проекту SAP. Тематика цих курсів варіюється від загального огляду системи до ретельного вивчення тієї чи іншої теми. Курси навчання діляться на три рівні складності:

Рівень 1 – Одно-дводенні курси, знайомство з технологією R/3 та методологією ASAP.

Рівень 2 – Три-п'ятиденні курси, що забезпечують початкову спеціалізацію в тій чи іншій області.

Рівень 3 – Три-п'ятиденні курси, що забезпечують глибокі пізнання в області, яка вивчалася на рівні 2.

Курси 1-го рівня призначені для тих, хто приймає важливі рішення по системі, ці курси рекомендується проходити до початку проекту впровадження.

Платформи та синхронізація. SAP NetWeaver є інтегрованою технологічною обчислювальною платформою SAP та технічною основою для багатьох додатків SAP, починаючи з SAP Business Suite. SAP NetWeaver позиціонується як сервіс-орієнтована прикладна та інтеграційна платформа. SAP NetWeaver надає середовище розробки і середовища виконання для додатків SAP та може використовуватися для індивідуальної розробки та інтеграції з іншими додатками і системами [3].

Інтеграція з іншими інструментами. SAP NetWeaver побудований в основному на мові програмування ABAP, але також використовує C, C++ та Java EE. SAP NetWeaver використовує відкриті стандарти та галузеві стандарти де-факто та може розширюватися за допомогою таких технологій, як Microsoft.NET, Java EE та IBM WebSphere, а також взаємодіяти з ними [3].

Загальний висновок. Таким чином, необхідно відзначити, що за рахунок збудованої методології управління проектами впровадження своїх програмних продуктів, що поєднується з програмним інструментом для її практичного застосування, SAP сприяє структуруванню великої кількості завдань, які необхідно виконувати на кожному етапі.

Серед інших переваг даної методології – зменшення часу між інсталяцією та запуском системи в виробництво у режимі реального часу, а також створення загальної моделі управління проектом, яка може згодом застосовуватися і для інших впроваджуваних компанією продуктів або оновлень системи.

У цілому ж корпоративна методологія ASAP пропонує досить вдалий комплексний інструментарій для управління проектами, що поєднує технологічну та управлінську частини у вигляді трьох основних компонентів:

синхронізація бізнес-пріоритетів клієнта (у залежності від індустрії) з конкретними рішеннями SAP через галузеві карти бізнес-цінностей (value maps) інструменту Solution Explorer;

карти управління проектами (маршрутні карти) з основними етапами та активностями за ними;

проектування, налаштування, тестування та експлуатація рішення за допомогою менеджера рішень SAP Solution Manager.

ЛІТЕРАТУРА

1. Вікіпедія. SAP SE. Електронний ресурс. Режим доступу: https://uk.wikipedia.org/wiki/SAP_SE
2. <https://blogs.sap.com/2013/09/17/the-all-new-asap-8-methodology/>
3. "NetWeaver Open Integration Platform".
<https://www.sdn.sap.com/irj/sdn/developerareas/netweave>.

Берестов Д.С., к.т.н., асистент кафедри

кафедра Програмних систем і технологій

Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Курченко О.А., к.т.н., доц.,

доцент кафедри Інформаційної та кібернетичної безпеки Державного університету телекомунікацій;

Покотило О.І., к.істор.н.,

керівник гол. деп. з питань нац.безп. та оборони Адміністрації президента України;

Особливості розробки програмного забезпечення для забезпечення захисту інформації від несанкціонованого доступу

Замовниками програмного забезпечення можуть бути як недержавні, так й державні установи. У випадку недержавних установ інформація, що циркулює в системі вимагає захисту комерційної тайни. Її несанкціонований виток може вести до втрати грошей або, навіть, бізнесу. Програмне забезпечення державних установ в більшості випадків може вимагати захисту державної тайни, за розголошення якої, згідно законодавства, встановлена відповідальність від адміністративної до кримінальної.

В більшості випадків розробники, які не мали досвіду створення програмного забезпечення для роботи з інформацією обмеженого доступу, концентрують всі свої зусилля виключно на функціональності програмного забезпечення, вважаючи, що захист інформації можна буде додати потім, як опцію. Насправді, ефективний захист інформації вимагає створення відповідної архітектури і використання відповідних методичних підходів, щодо створення програмного забезпечення. Для цього потрібно вже на перших етапах проектування враховувати основні вимоги щодо захисту інформації в інформаційних системах.

Забезпечення безпеки інформації неперервний процес, який полягає в обґрунтуванні і реалізації найбільш раціональних методів, способів і шляхів удосконалення та розвитку системи захисту, неперервному контролю її стану, виявленні її вузьких і слабких місць, а також протиправних дій. Безпека інформації може бути забезпечена лише при комплексному використанні всього арсеналу наявних засобів захисту в усіх структурних елементах виробничої системи і на всіх етапах технологічного циклу обробки інформації. Найбільший ефект досягається тоді, коли всі використовувані засоби, методи і заходи поєднуються в єдиний цілісний механізм – систему захисту інформації (СЗІ). При цьому функціонування системи повинно контролюватися, обновлятися і доповнюватися залежно від зміни зовнішніх і внутрішніх умов.

СЗІ розглядається як єдина сукупність правових, організаційних, фізичних заходів, програмно-технічних засобів, спрямованих на нейтралізацію загроз інформаційній безпеці. Постановка задачі розробки СЗІ формулюється як „різними засобами та заходами запобігти впливу загроз і відповідного витоку, руйнування та викривлення інформації до гарантованого рівня при мінімальних витратах”. При цьому необхідно враховувати, що незахищеність будь-якої ланки в структурі програмної системи може призвести до суттєвого знецінювання витрат, які були вкладені в здійснення відповідних заходів захисту інформації в усіх інших ланках.

Захищають і охороняють, як правило, не всю або не будь-яку інформацію, а найбільш важливу для власника, обмеження поширення якої приносить йому якусь користь або прибуток, можливість ефективно вирішувати завдання, що стоять перед ним.

Власниками інформації, яку захищають, можуть бути: держава та її структури (органи); підприємства, товариства, акціонерні товариства (у тому числі спільні); громадські організації; громадяни.

Конфіденційна або секретна інформація поділяється на такі види: державна таємниця; комерційна таємниця; банківська таємниця; податкова таємниця; службова таємниця; професійна таємниця; персональні дані.

Засекречування інформації – це сукупність організаційно-правових заходів, регламентованих законами й іншими нормативними актами, щодо введення обмежень на поширення й використання інформації в інтересах її власника. Інформація повинна

залишатися секретною або конфіденційною доти, поки цього вимагають інтереси національної безпеки або комерційної й комерційної діяльності підприємства.

Розсекреченню (розголошенню) не підлягають відомості, що стосуються особистого (неслужбового) життя громадян країни, якщо на інше немає згоди самих громадян, а у випадку їхньої смерті - їх найближчих родичів. Інший порядок такого розсекречення розглядається через суд.

Будь-яка фірма, займаючись своєю діяльністю, функціонує в деякому просторі (зовнішньому середовищі), і на її розвиток впливає ряд факторів (позитивних або негативних) як зовнішніх, так і внутрішніх. Негативні фактори часто називають факторами загроз або **загрозами**. Джерелами **зовнішніх** загроз є: конкуренти; кримінальні структури; корумповані елементи держаних структур; природні катаклізми і техногенні катастрофи. **Внутрішні** загрози створюють: засновники та вище керівництво; менеджери різних рівнів; персонал.

Об'єктами загроз є відомості про склад, стан і діяльність об'єкта захисту (персоналу, матеріальних і фінансових цінностей, інформаційних ресурсів).

Загрози інформації полягають у порушенні її цілісності, конфіденційності, повноти і доступності.

Джерела загроз мають на меті ознайомлення з відомостями, що охороняються, їх модифікацію в корисливих цілях і знищення для нанесення прямого матеріального збитку.

Небезпечність загроз визначається їх впливом на порушення конфіденційності, цілісності, доступності, спостереженості інформації, наслідками чого можуть бути: збитки, привнесені галузі діяльності, до якої відноситься інформація, через погіршення умов інформаційного забезпечення діяльності; збитки, привнесені автоматизованій системі.

Оцінка небезпечності загроз за рівнем очікуємих збитків у разі їх реалізації свідчить, що рівень такої небезпечності в першу чергу залежить від ступеня секретності інформації, що обробляється. Чим вище цей ступінь, тим небезпечніші наслідки від реалізації загроз.

Основними напрямками захисту інформації є правовий, організаційний і інженерно-технічний захисту інформації як складові комплексного підходу до забезпечення інформаційної безпеки.

Захист інформації може бути здійснений за допомогою таких засобів: фізичних; апаратних; програмних; криптографічних; стеганографічних.

До **способів захисту** належать усілякі заходи, шляхи, способи і дії, що забезпечують попередження протиправних дій, їхнє запобігання, припинення і протидію несанкціонованому доступу.

Концепція забезпечення захисту інформації є основним правовим документом, що визначає захищеність підприємства від внутрішніх і зовнішніх загроз.

Заходи захисту інформації від несанкціонованого доступу, вимоги до реалізації функціонального профілю та гарантії захищеності, послуги, які відповідають профілю захищеності, а також засоби та заходи, що забезпечують реалізацію послуг, засоби і заходи запобігання витоку інформації технічними каналами визначаються у **Політиці безпеки** інформації.

Вимоги до захисту інформації реалізуються послугами, які включають технічні, апаратно-програмні та організаційні заходи. Перелік необхідних послуг, які повинна реалізовувати СЗІ, складає функціональний профіль захищеності інформації. Порядок вибору функціональних профілів захищеності інформації від несанкціонованого доступу у залежності від характеристик оброблюваної інформації викладено у [2].

Концепція повинна визначати порядок оцінки достатності заходів та засобів створення СЗІ.

Вимоги до рівня забезпечення захисту інформації визначаються співвідношеннями між збитками, які очікуються внаслідок реалізації загроз і витратами на створення СЗІ. Розрахунки свідчать, що рівень збитків від впливу загроз на інформацію, яка відноситься до державної таємниці, незрівнянно вище витрат на створення СЗІ, тому така інформація повинна захищатися з імовірністю близькою до одиниці. Відступ від такої вимоги може бути у випадках, коли інформація відноситься до конфіденційної і збитки від реалізації загроз можуть бути порівняні з витратами на створення СЗІ.

Конкретизація заходів і засобів захисту інформації повинна проводитися при реалізації вимог, викладених у технічному завданні на створення СЗІ в конкретних за призначенням автоматизованих/програмних системах.

Викладені положення стосуються інформації з обмеженим доступом, що відносяться до конфіденційної або такої, що становить державну таємницю. Але відповідно до [5] захисту в системі підлягає також відкрита інформація, яка є власністю держави і використовується для забезпечення діяльності державних органів або органів місцевого самоврядування, а також інформація про діяльність зазначених органів, яка оприлюднюється в Інтернет, інших глобальних телекомунікаційних мережах і системах.

Зазначена інформація під час обробки в автоматизованій/програмній системі повинна зберігати цілісність, що забезпечується захистом її від несанкціонованої модифікації або знищення, а також повинна забезпечуватися її доступність для ознайомлення усім користувачам.

Виконання цих вимог повинно забезпечуватися відповідно до технічного завдання на створення автоматизованої/програмної системи або окремого технічного завдання на створення системи захисту інформації.

ЛІТЕРАТУРА

1. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. Затверджено наказом ДСТСЗІ СБ України від 28.04.1999 №22.
2. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблювальної інформації від несанкціонованого доступу. Затверджено наказом ДСТСЗІ СБ України від 28.04.1999 №22.
3. ТР ЕОТ-95. Тимчасові рекомендації з технічного захисту інформації у засобах обчислювальної техніки, автоматизованих системах і мережах від витоку каналами побічних електромагнітних випромінювань і наводок. Затверджені наказом ДСТЗІ від 09.06.1995 №25.
4. ТР ТЗІ-ПЕМВН-95. Тимчасові рекомендації з технічного захисту інформації від витоку каналами побічних електромагнітних випромінювань і наводок. Затверджені наказом ДСТЗІ від 09.06.1995 №25.
5. Правила забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах. затверджені постановою КМУ від 29.03.06р. №373.

Берестов Д.С., к.т.н., асистент кафедри
кафедра Програмних систем і технологій

Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Курченко О.А., к.т.н., доц.,

доцент кафедри Інформаційної та кібернетичної безпеки Державного університету телекомунікацій;

Покотило О.І., к.істор.н.,

керівник гол. деп. з питань нац.безп. та оборони Адміністрації президента України;

Побудова комплексної системи захисту інформації в автоматизованих системах

Вимоги щодо захисту інформації мають бути враховані вже на етапі проектування та створення програмного забезпечення, яке входить до складу автоматизованої системи (АС). Але на цьому робота щодо захисту не закінчується. Наступною складовою має бути створення комплексної системи захисту інформації (КСЗІ). Якщо технічні вимоги щодо захисту інформації не були враховані на етапі проектування та створення АС, то створити КСЗІ скоріше всього буде неможливо або вкрай важко. Якщо технічні вимоги були враховані, то далі починається етап створення КСЗІ із залученням інших нетехнічних галузей діяльності.

Комплексна система захисту інформації розглядається як єдина сукупність правових, організаційних, фізичних заходів, програмно-технічних засобів, спрямованих на нейтралізацію загроз інформаційній безпеці. Постановка задачі розробки КСЗІ формується як «різними засобами і заходами запобігти впливу загроз і відповідного витоку, руйнування та викривлення інформації до граничного рівня при мінімальних витратах». При цьому необхідно враховувати, що незахищеність будь-якої ланки в структурі автоматизованої системи (АС) може призвести до суттєвого знецінювання витрат, які були покладені на здійснення відповідних заходів захисту інформації в усіх інших ланках.

Оскільки КСЗІ є складовою частиною АС, терміни початку, завершення робіт, етапи створення КСЗІ повинні співпадати з розробкою самої АС, замовник АС повинен виступати і замовником КСЗІ. Розробник АС при необхідності та за погодженням з замовником для виконання робіт зі створення КСЗІ може залучати інші організації (співвиконавців), які мають необхідні ліцензії (дозволи) на проведення таких робіт. Такий підхід до створення КСЗІ забезпечує захист інформації на всіх етапах життєвого циклу АС.

Етапи створення КСЗІ включають:

1. **Обґрунтування необхідності** створення КСЗІ. Підставою для визначення необхідності створення КСЗІ є вимоги чинного законодавства, що встановлюють обов'язковість обмеженого доступу до інформації, що обробляється в АС.

2. **Обстеження середовищ** функціонування об'єктів інформаційної діяльності (ОІД).

Обстеження ОІД проводиться після категорювання його відповідно до [9]. Метою обстеження ОІД є вивчення інформаційної діяльності, визначення об'єктів захисту, аналіз та побудова моделі загроз. Докладно порядок проведення обстеження викладено в [7]. Результати проведення робіт з обстеження оформлюються актом, який підписується членами комісії та затверджується керівником установи/закладу.

За результатами обстеження розробляється **модель загроз та порушників** відповідно до положень [1, 6, 10].

3. Розробка **політики безпеки** інформації в АС являє собою набір вимог, правил, обмежень, рекомендацій, які регламентують порядок обробки інформації і спрямовані на захист її від потенційних загроз. Розроблення політики безпеки передбачає проведення наступних робіт: визначення інформаційних та технічних ресурсів в АС, що підлягають захисту; уточнення моделі загроз і порушників, наліз ризиків, вибір основних рішень з протидії можливим загрозам, формування загальних вимог, правил, обмежень, рекомендацій, які регламентують використання окремих засобів і заходів захисту інформації, правила діяльності користувачів усіх категорій. Вимоги до формування

політики безпеки та її змісту визначені відповідно з положеннями [1, 6]. Політика безпеки є основним документом, який визначає вимоги до пунктів технічного завдання на створення КСЗІ, оформляється у вигляді окремого документа Плану захисту.

4. Розробка **технічного завдання** на створення КСЗІ.

Технічне завдання на створення КСЗІ оформляється у вигляді окремого (часткового) ТЗ, доповнення до загального ТЗ на АС або розділу до загального ТЗ на АС. В ТЗ на створення КСЗІ викладаються вимоги до функціонального складу і порядку розробки і впровадження засобів і заходів захисту інформації. ТЗ повинно містити наступні розділи:

1) Загальні відомості. В цьому розділі вказуються структурна схема і склад АС, характеристика інформації (ступінь її обмеженості), характеристика фізичного середовища (територіальне розміщення компонентів АС), наявність категоризованих приміщень, характеристики каналів зв'язку (пропускна спроможність, типи кабельних ліній, види зв'язку з віддаленими сегментами АС і користувачами, характеристика персоналу (кількість і категорії за формою допуску користувачів), особливості функціонування АС (режими роботи), реалізовані організаційні та фізичні заходи захисту, потенційні загрози інформації (можливі способи НСД, технічні канали витоку інформації);

2) Мета і призначення КСЗІ. Визначається функціональне призначення КСЗІ, нормативно-правові документи, що регламентують організацію захисту інформації в АС.

3) вимоги до КСЗІ.

У частині захисту від несанкціонованого допуску (НСД). Захист від НСД повинен гарантувати виконання вимог конфіденційності, цілісності, доступності і спостереженості ІЗОД.

Вимоги до захисту інформації реалізовується послугами, які включають технічні, апаратно-програмні та організаційні заходи. Перелік необхідних послуг, які повинна реалізувати КСЗІ, складає відповідний функціональний профіль захищеності. Зміст функціональних профілів захищеності визначається вимогами захисту певної інформації від певних загроз і відомих функціональних послуг, що дозволяють протистояти можливим загрозам. Порядок вибору функціональних профілів захищеності від ступеня обмеженості інформації викладено у [2].

Як засоби і заходи інформації від НСД повинні розглядатися: а) **програмно-технічного характеру**, б) **організаційного характеру**.

4) вимоги до **складу проектної та експлуатаційної документації**. Перелік необхідної проектної та експлуатаційної документації визначається розробником КСЗІ і погоджується з замовником. В цілому при розробці ТЗ на створення КСЗІ необхідно керуватись НД [3].

5. На підставі і у відповідності з ТЗ на створення КСЗІ розробником КСЗІ **обґрунтовуються і приймаються проектні рішення** щодо реалізації вимог ТЗ.

Проектна документація складається з «Ескізного», «Технічного» та «Робочого проектів КСЗІ». В зазначених проектах або в «Техноробочому проекті КСЗІ» розробляються проектні рішення, необхідні для реалізації вимог ТЗ на КСЗІ, структури КСЗІ (організаційної, технічної і програмних засобів), рішень щодо архітектури комплексу засобів захисту, механізмів реалізації визначених функціональним профілем послуг захисту інформації. Здійснюється розробка, оформлення та затвердження робочої та експлуатаційної документації, передбаченої ТЗ на КСЗІ. Робоча документація повинна містити детальні рішення щодо реалізації технічних вимог, тестування, проведення пусконаладжувальних робіт, випробувань КСЗІ.

6. **Випробування** КСЗІ.

Випробування реалізованого проекту КСЗІ розподіляється на попередні випробування, дослідну експлуатацію та проведення державної експертизи.

Метою попередніх випробувань є перевірка працездатності, КСЗІ та перевірка можливості прийняття її у дослідну експлуатацію. Попередні випробування проводяться за програмою і методиками, які готуються розробником КСЗІ і узгоджуються із замовником. Для проведення випробувань замовником КСЗІ створюється комісія, головою комісії призначається представник замовника. Результати попередніх випробувань оформлюються «протоколом випробувань», де вказуються недоліки, виявлені у ході випробувань, заходи та терміни їх удосконалення. Після усунення недоліків та

відповідного коригування проектної та експлуатаційної документації оформлюється акт про прийняття КСЗІ у дослідну експлуатацію.

Під час дослідної експлуатації доопрацьовується (за необхідністю) програмне забезпечення, експлуатаційна документація. За результатами дослідної експлуатації складається акт щодо можливості (неможливості) проведення державної експертизи КСЗІ.

Державна експертиза КСЗІ проводиться з метою визначення відповідності її вимогам ТЗ на її створення, можливості передання її в експлуатацію. Порядок проведення державної експертизи викладений у [8]. Організація виконавець експертизи за результатами аналізу наданих матеріалів формує програму та окремі методики проведення експертизи. Програма проведення експертизи погоджується із замовником та уповноваженим органом, окремі методики - з уповноваженим органом. Результати роботи оформлюються у вигляді протоколів за підписом експертів – виконавців роботи. Організація-виконавець експертизи складає експертний висновок, у разі затвердження якого уповноваженим органом організації-виконавцю надається зареєстрований України Атестат відповідності.

ЛІТЕРАТУРА

1. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. Затверджено наказом ДСТСЗІ СБ України від 28.04.1999 №22.
2. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу. Затверджено наказом ДСТСЗІ СБ України від 28.04.1999 №22.
3. НД ТЗІ 3.7-001-99. Методичні вказівки щодо розробки технічного завдання та створення комплексної системи захисту інформації в автоматизованій системі. Затверджено наказом ДСТСЗІ СБ України від 28.04.1999 №22.
4. ТР ЕОТ-95. Тимчасові рекомендації з технічного захисту інформації у засобах обчислювальної техніки, автоматизованих системах і мережах від витоку каналами побічних електромагнітних випромінювань і наводок. Затверджені наказом ДСТЗІ від 09.06.1995 №25.
5. ТР ТЗІ-ПЕМВН-95. Тимчасові рекомендації з технічного захисту інформації від витоку каналами побічних електромагнітних випромінювань і наводок. Затверджені наказом ДСТЗІ від 09.06.1995 №25.
6. НД ТЗІ 1.4-001-2000. Типове положення про службу захисту інформації в автоматизованій системі. Затверджено наказом ДСТСЗІ СБ України від 04.12.20000 №53.
7. НД ТЗІ 3.7-003-05. Порядок проведення робіт зі створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі.
8. Положення про державну експертизу в сфері технічного захисту інформації, затвердженого наказом ДСТСЗУ СБ України від 29.12.1999 №62.
9. Тимчасового положення про категорювання об'єктів (ТПКО-95), затвердженого ДСТЗІ України 10.07.1995р. № 35
10. НД ТЗІ 3.7-003-05. Порядок проведення робіт із створення комплексної системи захисту інформації в інформаційно-телекомунікаційній системі

Бражиненко Максим, студент магістратури
Ляшуга Микола,
 співробітник компанії **Walkout Technologies GmbH**;
 студент 4 курсу бакалаврату,
Самойленко Олег,
 співробітник компанії **GlobalLogic**;
 студент 4 курсу бакалаврату,
Дронь Дмитро,
 співробітник компанії **Stearling Web Agency**;
 студент 4 курсу бакалаврату,
Цехмейструк Андрій,
 співробітник компанії **УКРСІББАНК**;
 студент 4 курсу бакалаврату,
Шаламай Єлизавета,
 співробітник компанії **Epam Systems**;
 студент 4 курсу бакалаврату,
 кафедра Програмних систем і технологій
 Факультет інформаційних технологій
 Київський національний університет імені Тараса Шевченка, м. Київ, Україна
Беляченко О.В.,
 комерційний директор компанії **Софт-Рейтинг Консалт**.

Аналіз відповідності навчальних дисциплін вимогам виробництва програмного забезпечення

Індустрія інформаційних технологій є однією з найбільш динамічно змінюючихся. Фреймворки, платформи, технології, пакети старіють та виходять із підтримки майже щороку. Це актуалізує проблему побудови навчального плану у відповідності зі змінами та тенденціями індустрії. Слід також зазначити що окрім безпосередньо технологій еволюціонують також процеси і якщо в минулому, ще 10 років тому назад, компанії розраховували та шукали «універсальних солдатів» то зараз частіше за все фахівці мають доволі вузьку спеціалізацію і доцільно зазначити, що тенденція дріблення буде мати місце і надалі, а потреба «універсальних» навичок буде зникати. Важливо зауважити, що ми зараз розглядаємо відповідність знань і навичок випускника ВНЗ із позиціями «молодшого розробника програмного забезпечення», «тестувальника» (мовою професійного сленгу «Junior»).

Кожного разу коли випускник ВНЗ починає професійну діяльність і відкриває вихідний код свого першого проекту, він стикається з припущенням, що йому відомо, чому світ такий як він є. А його керівник з нерозумінням, чому його навчали у ВНЗ через те що в його підлеглого є знання і навички про, що завгодно крім того, що потрібно. А отже метою цієї роботи є узагальнення не відповідностей, їх визначення та пошук можливих шляхів вирішення:

Зв'язок між дисциплінами (залежності) – навіть швидкий огляд навчальних дисциплін демонструє, що зв'язки між ними не встановлені та їх черговість викладання створює штучні перешкоди для студента ВНЗ. Найяскравішим прикладом є «Архітектура програмного забезпечення» на 2 курсі. Очевидно, що Архітектура тісно пов'язана з управліннями проектів, роботою з вимогами, вимагає знання хоча б поверхневого більшості технологічних доменів (Хмари, Бази Даних, Веб технології) і т.і а тому її викладання на 2 курсі у тому вигляді в якому вона існує на виробництві не можливе, через брак знань у студентів.

Зв'язок між спеціалізаціями – як вже було зазначено вище від випускника ВНЗ очікується вузька спеціалізація, але швидкість дріблення не дозволяє ВНЗ вчасно адаптуватися. Отже виникає проблема відновлення зв'язку між спеціалізаціями та отримання навичок, що пом'якшать перехід між ними. Більшість сучасних проектів імплементують безперервну доставку та інтеграцію (через потребу робити це швидше ніж конкуренти), а інструментарій реалізації змінюється повільно. Є ті хто називає це окремим

напрямок «DevOps» - що є частково правдою. Встановлення зв'язку між дисциплінами і більш вузькими спеціалізаціями можливе за рахунок часткового вивчення «DevOps» практик та інструментів, бо не залежно від мови програмування та специфіки професійної діяльності, ця частини залишиться такою як є, до деякої межі. Серед рекомендованих технологій слід зазначити наступні – Docker, Linux (адміністрування та мережі), Jenkins (або TFS), Groovy, Powershell, Bash, JavaScript інструменти автоматизації (наприклад Grunt, Gulp, Babel), Jira (або аналоги). Як результат випускники ВНЗ після закінчення навчання повинні отримати навички побудови додатку з автоматизованою перевіркою якості та розгортанням, отримати знання зі створення систем з безперервною доставкою та інтеграцією.

Зв'язок з практикою – є багато суто практичних речей які жодним чином не запам'ятаєш, бо правил/формул не існує. Є дослідження, але аналіз останніх вимагає достатньо багато часу, якого може й не бути. В індустрії інформаційних технологій вимоги є одним з найслабкіших місць для розробників, і їх розуміння довгий час залишається достатньо «туманним». Основною проблемою є побудова правильного ланцюга Рис 1.

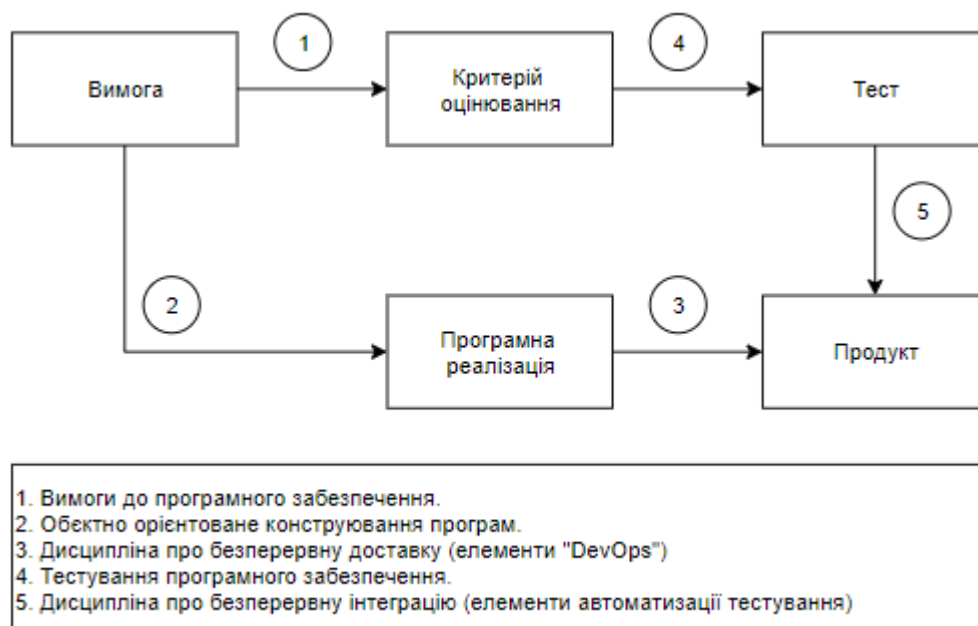


Рис.1 Ланцюг зв'язку вимоги та продукту.

Зв'язки між елементами розглянуті кожний окремо, на різних дисциплінах, але немає такої яка б ставила за мету узагальнення. Також можна відслідковувати наслідки проблеми відсутності зв'язку між дисциплінами, порядок розгляду з'єднань змішано в навчальному плані, є й такі що відсутні.

Висновок. Після детального аналізу поточного плану було зроблено висновок, що його необхідно змінити а саме відновити зв'язок між дисциплінами шляхом зміни їх черговості, додати елементи «DevOps» та автоматизації з метою більш гнучкого переходу між сміжними спеціалізаціями а також необхідністю додати узагальнення отриманих навичок.

Nesterenko O.V., Ph.D., Associate Professor, Leading researcher

Netesin I.E., Ph.D., Leading researcher

Polischuk V.B., Ph.D., Director

State Enterprise "Ukrainian Scientific Center for Information Technology Development", Kyiv, Ukraine, info@rit.org.ua

Information-analytical system for atmospheric air quality territorial management

According to World Health Organization air pollution is one of the major risk factors for respiratory and cardiovascular diseases. Ratified by Ukraine international documents require assessment of the effects air pollution on public health. Therefore, these issues are in the area of constant state attention. In Ukraine studies on assessment of air pollution by industrial enterprises and motor transport are conducting by various agencies.

However, their results are fragmented and disparate. This does not provide a comprehensive picture of the deterioration population health, which living in conditions of increased inhalation risk. At the same time, examples of successful solutions to this issue at the local and regional levels may differ significantly. This will create problems during formation of the national system by the Cabinet of Ministers of Ukraine Decree No. 827 of August 14, 2019 "Some Issues of State Monitoring in the Field of Atmospheric Air Protection" envisaged.

Proceeding from this, the development of scientifically sound recommendations and software-technical means for assessment and characteristics of atmospheric air on basis of integral indicators are necessary to provide population information and management decisions make in this field [1].

One of the ways to solve these problems is to create an information-analytical system (IAS), which should ensure information on the sources of emissions characteristics and air pollution state in order to assess inhalation risk to public health collection, processing, storage and analysis.

The IAS provides:

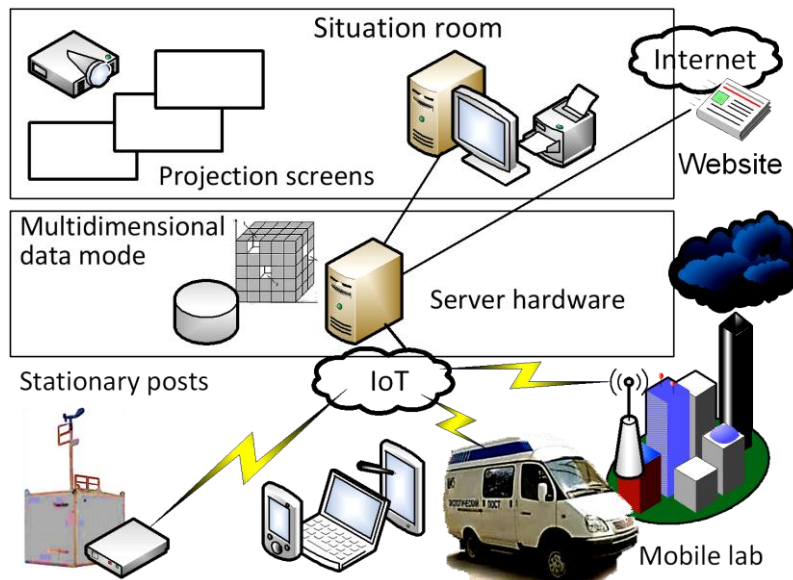
- automation of data collection on the characteristics and parameters of sources of atmospheric emissions by collecting data from stationary observation points and mobile medical-ecological laboratories (including those available at the AM Marzeev Institute of Public Health of the National Academy of Medical Sciences of Ukraine), which a providing information on concentrations in atmosphere of such ingredients as particulate matter (PM₁₀, PM_{2.5}), carbon monoxide, nitric oxide, ozone, etc. [2];
- creation of a data warehouse and filling it with certain data on the objects of certification (from documents by enterprises to the department of territorial management body submitted for obtaining a permit for the emission of pollutants into the atmospheric air, in which the volumes of pollutants into air are justified);
- accumulated data analytical processing in order to determine the exceedance of hazard thresholds and possible impact this pollution on the population health and life;
- analytical data visual presentation on indicator panels to support decision-making in the situation room mode;
- transfer of necessary data to the territorial air quality management body and to the Ministry of Energy and Environmental Protection of Ukraine.

Architecture IAS is shown in the figure.

The basis of data analysis and research should be a formalized description (model) assessing state of atmospheric air pollution in relation to impact on population health and life. For this purpose general characteristic indicators of sources pollution (industrial enterprises and vehicles), expositions (pollutants concentrations in the air) and risk magnitudes to public health are used. Model on basis of ontological approach is formed [3].

To ensure the accumulation of transactional and analytical information and its multidimensional analysis, data warehouse architecture is being developed.

In order to evaluate (select) alternatives for determining emergency response measures related to exceeding the maximum permissible concentrations, it is advisable the expert evaluation methods to use [4].



The system important element is design and visualization on situation room indicators panel of key assessment indicator of atmospheric air pollution state in relation to population health and vital effects for support decision-making, separately by displaying spatial data using the geographic information system [5], as well as on a dedicated website.

On-line data send from stationary observation points and mobile medical-ecological laboratory on the basis Internet of Things (IoT) technology is

realized.

Thus, the IAS is a component of the resource support of monitoring subjects with modern information technologies and equipment. The system will adequately identify the risk of environmental pollution, quantify its impact on public health and develop recommendations for measures to reduce environmental and hygiene risks. This will help reduce the socio-economic losses associated with population mortality and morbidity risk and other indirect losses.

References

1. Turos O., Maremukha T., Petrosian A., Mykhina L., Morhulova V. Integral indicators as tools for air quality assessment // Довкілля та здоров'я, № 1 (90), 2019. – С. 51-55.
2. Turos O.I., Maremukha T.P., Petrosyan A.A., Brezits'ka N.V. Doslidzhennya zabrudnennya atmosferного povitrya z vazhenymy chastynkami pylu (RM10 ta RM2,5) u m. Kyievi // Dovkilliya ta zdorov'ya. 2018. №4(89). S. 36-40.
3. O. Nesterenko, O. Trofymchuk. Patterns in forming the ontology-based environment of information-analytical activity in administrative management // Eastern-European Journal of Enterprise Technologies, 2019, № 5/2 (101). – P. 33-42. DOI: 10.15587/1729-4061.2019.180107.
4. Nesterenko O.V., Netesin I.YE., Polishchuk V.B. Intehratsiya metodu analizu iyerarkhiy, komp'yuternykh ontolohiy ta oriyentovanykh hrafov dlya ekspertnoho otsynyuvannya u zadachakh bahatokryterial'noho vyboru // «Suchasni problemy informatyky v upravlinni, ekonomitsi, osviti ta podolanni naslidkiv Chornobyl's'koyi katastrofy» : [materialy XVIII mizhnarodnoho naukovo seminaru] / za nauk. red. d.e.n., prof. M. M. Yermoshenka. – K.: Natsional'na akademiya upravlinnya, 2019. – S. 8-10.
5. Bondarchuk S.V., Zakalad M.A., Netesin I.YE., Polishchuk V.B. Metodychni pidkhody do pobudovy paneli indykatoriv prohramno-tekhnichnoho kompleksu pidtrymky pryynyattya rishen' u sferi viys'kovoyi lohistyky // Zbirnyk materialiv naukovopraktychnoho seminaru «Metodolohichni pytannya informatyzatsiyi lohistyky v upravlinni oboronnykh resursamy» (9 hrudnya 2016 r., m. Kyiv). – Kyiv: UkrNTS RIT. – S. 47-52.

Артеменко Є.В., Мідоян Е. Р. студенти 3 курсу
кафедра Програмних систем і технологій
Факультет інформаційних технологій
Київський національний університет імені Тараса Шевченка, м. Київ, Україна

Обробка даних комп'ютерним зором в інформаційному просторі за допомогою лінійної фільтрації

З настанням періоду індустріалізації людина змогла збільшити свою силу. Те що за робочий день раніше виконували 100 людей, тепер робить машина, за якою наглядає й обслуговує одна людина. Але з розвитком технології «комп'ютерний (технічний) зір» навіть присутність такого спостерігача стає не обов'язковою.

Комп'ютерний зір – це технологія, що спрямована на знаходження об'єктів, їх аналіз й класифікацію. Спектр використання цього напрямку настільки широкий, що вже сьогодні ми зустрічаємось з ним кожен день: розблокування смартфона лицем, автономні транспортні засоби, які не потребують участі водія в подорожі, пошук зображень в інтернеті. Хоча бувають й менш благородні способи його використання, як Netflix, не раз спійманий на заміні кіноафіш на схожі з тими, що вже сподобалися користувачу. Що вже казати про медицину, в якій машина може поставити діагноз майже без участі доктора або про архітектуру, в якій 3D моделі споруд збираються з фото.

На цьому моменті потрібно провести межу між комп'ютерним зором й машинним. Комп'ютерний зір дозволяє машині бачити й аналізувати, а машинний зір є областю використання в промисловості.

Серед проблематики комп'ютерного зору виділяють наступні:

- розпізнавання – розпізнання декількох гарно вивчених об'єктів;
- ідентифікація – знаходження особливих речей, таких як лице людини;
- виявлення – знаходження й виявлення критеріїв й умов, зазвичай для простих об'єктів;
- відновлення 3D форми по 2D зображенням – створення однієї моделі з декількох зображень, часто використовується архітекторами, археологами або діячами мистецтва;
- відновлення зображень – виявляє та фрагментує зображення й за допомогою алгоритмів й машинного навчання дозволяє не тільки відновити, але й в деяких випадках покращити якість зображення;
- Розпізнавання тексту – розпізнання та відтворення тексту, що допомагає в перенесенні документів.

Для вирішення таких задач використовуються різні методи обробки зображення. Одним з таким методів є лінійна фільтрація, який часто можна зустріти в електроніці, цифровій обробці сигналів і зображень, в оптиці, теорії управління. В комп'ютерному зорі він використовується для видалення шумів, пошуку меж й кутів.

Гарним прикладом лінійної віконної фільтрації в просторовій області є згортка, а точніше ковзне середнє у вікні. Результат фільтрації – це значення математичного очікування, обчислене за всіма пікселями вікна. Це призводить до загублення фокусу на зображенні й воно стає не таким чітким. А вже для визначення меж ми використовуємо згортку таким чином, щоб отримати горизонтальні та вертикальні, при накладанні яких вже можна отримати всі межі зображення. Цей метод вважається одним з простих, але ефективних. Кореляцію вже використовують для знаходження схожості між зображеннями й окремими їх елементами. Такий спосіб також вважається простим, особливо коли використовуються копії.

За допомогою таких інструментів в більшості пошукових систем є можливість розкласти зображення на речі, що знаходяться на ньому, розважальні та художні ресурси розпізнають плагіат, а відеохостинги здійснюють фільтрацію небажаного контенту. Гарним прикладом може слугувати Facebook, що дає змогу ідентифікувати людей по фото й тим самим спрощує життя користувачам.

Комп'ютерний зір є однією з найважливіших, але в той самий час однією з найбільш недооцінених технологій в Інтернеті, способи використання якої нам ще потрібно буде дослідити.

ЛІТЕРАТУРА

- Л. Шапиро, Дж. Стокман. Компьютерное зрение. — М.: Бином. Лаборатория знаний, 2006. — 752 с
С. Ю. Желтов. Машинное зрение. (МИРЭА). 2006 – 184 с

Карандась Д.Д. студент 3 курсу

Ткаченко М.В., к.т.н., асистент кафедри ПСТ

Федоренко Р.М., к.е.н., асистент кафедри ПСТ

кафедра Програмних систем і технологій

Факультет інформаційних технологій

Київський національний університет імені Тараса Шевченка, м. Київ, Україна

Використання нейронних мереж при оцінці якості ПЗ

Розвиток інформаційних технологій у сучасному світі визначив потребу у оцінці якості програмного забезпечення. Показник якості програмного забезпечення є важливим критерієм встановлення відповідності системи до певних вимог. Таким чином оцінювання системи передбачає велику кількість різних розрахунків що загалом формують цілісну оцінку системи. В цілому якісний показник програмного забезпечення є досить відносним та індивідуальним значенням, оскільки він залежить від потреб замовника та дій розробника системи. Залучення нейронних мереж у процес оцінки якості програмного забезпечення може забезпечити формування більш деталізованих та індивідуальних показників для різних систем за рахунок отриманого досвіду у оцінці інших систем. Використання інтернету може допомогти у накопичуванні бази оцінювання за допомогою оцінки великої кількості систем різними користувачами. Отже, це забезпечить швидкий аналіз даних і формування середнього фіксованого показнику оцінки програмного забезпечення.

Процес оцінки якості програмного забезпечення з використанням нейронних мереж

Процес оцінки якості програмного забезпечення передбачає отримання певних показників за наданим характеристичним набором значень системи. На даний момент в інтернеті розміщено купа інформації що стосується процесу оцінки програмного забезпечення, починаючи з визначення метрик системи і до показників більш високого рівня, таких як характеристичні набори значень або індивідуально сформовані критерії. Що стосується метрик, то до таких відносять: кількість рядків коду, цикломатична складність, кількість помилок коду, відповідність вимогам коду та інші. До індивідуальних показників можливо віднести такі критерії як: структурованість, ступінь логічного ділення коду, відповідність сучасним стандартам, зрозумілість коду та інші.

Кажучи про нейронні мережі, то це системи що мають здатність навчатися і формувати відповіді в залежності від досвіду. В основі нейронної мережі лежать нейрони та зв'язки між ним. Сам нейрон представляє значення або функцію що формує значення опираючись на результати роботи минулого шару і закладених даних. Таким чином вони можуть виступати як механізм що може обробляти набори даних і формувати результуючий показник оцінки системи.

Характеристики оцінювання якості програмного забезпечення

При оцінюванні якості програмного забезпечення важливо зрозуміти кожну сторону для яких ця оцінка формується. Розробник має подбати щоб система мала як технічну повноту так і відповідала вимогам звичайних користувачів. Цей поділ дозволить нейронній

мережі формувати не тільки конкретні результати, але і комплекси уподібнених значень, тобто, елементів які можуть бути притаманні певній групі даних. У інтернеті та інших джерелах інформації визначена велика кількість стандартів що визначають якісні характеристики оцінювання систем, як приклад можемо взяти стандарт ISO/IEC 9126.

Стандарт ISO/IEC 9126 встановлює такі характеристики:

1. Функціональність — здатність системи вирішувати потрібні задачі користувача.
2. Надійність — можливість системи підтримувати працездатність у визначених умовах.
3. Зручність — здатність системи бути зручною у навчанні та використанні.
4. Продуктивність — можливість системи підтримувати працездатність з певним набором ресурсів.
5. Зручність супроводу — зручність супроводу системи.
6. Переносимість — здатність системи зберігати працездатність при перенесенні з одного оточення в інше.

Кожна з цих характеристик має свій комплекс метрик за якими можливо оцінювати системи та надавати їх для обробки нейронній мережі.

Роль нейронної мережі в оцінці якості програмного забезпечення

Нейронна мережа може виступати засобом отримання, обробки та формування результуючого показника або засобом ранжування показників по їх групам уподібнених характеристик і формування комплексу результатів. Також нейроні мережі є автоматизованим засобом тестування, тобто ми можемо проводити тестування готових систем за рахунок чого перевіримо можливості і технічні рамки програмного забезпечення. Цей підхід відрізняється тим, що він проводить перевірку системи безпосередньо під час взаємодії з нею. У свою чергу інтернет може стати середовищем що надає можливість користувачам та розробникам спробувати оцінити власне програмне забезпечення. Такі засоби мають покращити розробку програмного забезпечення та підвищити якість продукції що розробляється компаніями на сучасному ринку.

На даний момент популярність нейронних мереж стрімко зростає що сприяє їх розвитку і залучення у підтримку людської діяльності. Актуальність системи оцінки якості програмного забезпечення з використанням нейронних мереж є переконливою, оскільки зараз існує велика кількість засобів і додатків для аналізу, але вони не надають можливість аналізу програм написаних на різних мовах або не відповідають необхідній функціональній повноті для роботи з ними.

ЛІТЕРАТУРА

ISO/IEC 9126. Information Technology. – Software Quality Characteristics and metrics, 1997 – 7-11 с.

Холявкін Іван, студент, бакалаврату;
Ковалюк Т.В., к.т.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Автоматизація визначення переліку дисциплін за спеціальністю «Інженерія програмного забезпечення» на основі мотиваційного листа студента

Освіта є основою інтелектуального, духовного, фізичного і культурного розвитку особистості, її успішної соціалізації. Індивідуальна освітня траєкторія – це персональний шлях реалізації особистісного потенціалу здобувача освіти, що формується з урахуванням його здібностей, інтересів, потреб, мотивації, можливостей і досвіду, ґрунтується на виборі здобувачем освіти видів, форм і темпу здобуття освіти, суб'єктів освітньої діяльності та запропонованих ними освітніх програм, навчальних дисциплін і рівня їх складності, методів і засобів навчання. Індивідуальна освітня траєкторія в закладі освіти може бути реалізована через індивідуальний навчальний план [1]. Визначити особистість кандидата, почути про події, роздуми, переживання, які привели здобувача освіти до тої мрії чи бачення свого майбутнього. Для університетів важливо не просто набрати схильних до навчання студентів, а створити спільноту, у якій студенти зможуть заряджати своєю мотивацією інших, збагачуватися культурно, відчувати розуміння і підтримку. Оцінюючи мотиваційний лист, важливо визначити, чи може претендент збагатити процес навчання своїм унікальним і оригінальним трактуванням матеріалу чи предмету, чи він просто буде переповідати підручник.

Міністерство освіти і науки України розглядає необхідність включити до переліку документів, які подають абітурієнти до приймальних комісій ЗВО, мотиваційні листи. Отже, автоматизація обробки таких документів, які можуть бути основою для побудови персональних освітніх траєкторій навчання, є актуальною задачею.

Постановка задачі. Необхідно розробити програмне забезпечення для обробки тексту, написаного природною мовою, з метою побудови потрібного контексту. На основі аналізу мотиваційних листів потрібно визначити схильність та бажання абітурієнтів або студентів до навчання за певною спеціальністю та їх здатністю опановувати певні дисципліни. Кінцевою метою є побудова персональних освітніх траєкторій навчання, які стимулюватимуть студента до креативної, творчої роботи.

Опис методу розв'язання задачі. Програма виділяє речення з текстового файлу, з речення виділяє слова, визначає та видаляє стоп-слова, визначає частини мови, розпізнає та будує залежності між словами в реченні, будує дерева залежностей слів, розпізнає іменовані сутності речення, видобуває факти з текстового файлу та будує потрібний контекст [2].

Опис структури програми. В структурі програми виділені такі блоки: отримання ключових чанків, створення бази ключових чанків, створення списку дисциплін за базою, створення пріоритетів.

Чанк – це частина статичного тексту який можна вбудувати в шаблон, в інший чанк, або викликати в сніппеті. Чанк має ті ж властивості що і шаблон за винятком того, що не може бути призначений ресурсу безпосередньо. Чанк не може містити будь-якої виконуваний код, але в ньому можна викликати сніппети для виведення динамічного контенту.

Блок 1. Отримання ключових чанків.

Ядром цієї програми є використання NLP, яке в Python виражене через модулі spaCy і textacy. З їх допомогою можна отримати чанки, які представляють невеликі зв'язані частини тексту, які описують один об'єкт.

Блок 2 Створення бази ключових чанків.

Серед отриманих чанків є, як потрібна інформація, так і зайва. Далі задається список дисциплін. Для простоти використовуємо текстовий файл. Кожен абзац описує окрему дисципліну. Нехай дисципліну і ключові слова розділяє двокрапка, а ключові слова символ ^, бо ці символи не зустрічаються у самих чанках. Далі необхідно розробити функцію, яка просто дописує знайдений в мотиваційному листі ключовий чанк з

розділювачем у абзаци, які вказав користувач. Таким чином, після аналізу великої кількості мотиваційних листів програма отримує достатню базу для подальшої досить коректної роботи програми.

Блок 3. Створення списку дисциплін за базою чанків.

За тим самим алгоритмом отримуємо ключові чанки. Далі потрібно порівняти їх з базою. Саме для цього за допомогою розділювачів (перший - знак переносу, другий - двокрапка, третій - ^) ми перетворюємо текст файлу на тривимірний масив, де на першому рівні знаходяться блоки даних для кожної дисципліни, на другому (в цих блоках) - назва дисципліни та список ключових чанків, а на третьому (список ключових чанків) - в якості елементів виступають ключові слова). Тепер ми просто для кожного отриманого чанку з мотиваційного списку перевіряємо, чи є він хоча б в якомусь з цих списків, і якщо є, то записуємо назву дисципліни у множину дисциплін студента. Використання типу даних «множина» запобігає повторень ключових слів.

Також важливою складовою цього блоку є створення файлу студента, який міститиме дані, необхідні для визначення переліку дисциплін для кожного студента. За допомогою модуля `shelve` можна записати у цей файл отриманий список дисциплін.

Блок 4. Створення пріоритетів

Останнім кроком програми є створення механізму побудови списків пріоритетів. Список пріоритетів представляє із себе словник, де ключем є пріоритет, а значенням є дисципліна. На етапі введення значень, їх не можна дублювати, але можна вказувати будь-які дійсні числа, щоб мати можливість встановити пріоритет дисципліни. В кінці всі ці значення перетворюються на значення цілого вигляду.

Також додаємо можливість підписувати списки пріоритетів, щоб їх розпізнавати і мати можливість їх видалення, а список цих підписів можна зберігати окремо за ключем, яким є підпис. Зауважимо, що до кожного з підписів додається спеціальний постфікс, щоб список не міг замінити собою інший важливий елемент з ключем, ідентичним підпису.

Демонстрація роботи програми. Наступний скрін-шот (рис.1) демонструє фрагменти роботи програми на етапах визначення дисциплін та їх пріоритетів.

```
Current student (Guy Am I) information
Discipline list: Object Oriented Programming, Programming basics, Computer Graphics.
Press Enter to continue
1. Create disciplines list based on motivation letter
2. Create new priority list
3. Remove existing priority list
4. Change student's file
5. Output current information about the student
6. Quit to the program menu

Input an item: 2
Enter a priority for Object Oriented Programming: 4
Current list of priorities:
4.0 . Object Oriented Programming

Enter a priority for Programming basics: 3
Current list of priorities:
3.0 . Programming basics
4.0 . Object Oriented Programming

Enter a priority for Computer Graphics: 3.5
Current list of priorities:
3.0 . Programming basics
3.5 . Computer Graphics
4.0 . Object Oriented Programming

Final priorities list:
1 . Programming basics
2 . Computer Graphics
3 . Object Oriented Programming
Please, sign the priority list: Someone
```

Рис. 1. Фрагмент роботи програми визначення дисциплін за мотиваційним листом

Висновок. Розроблена програма обробки мотиваційних листів, що представлені текстами, написаними природною мовою з метою визначення уподобань студентів для побудови персональних освітніх траєкторій навчання.

Задача є актуальною, оскільки студентоцентрована парадигма вищої освіти є ключовим підходом до реформування системи освіти.

Програма написана мовою Python 3. Для реалізації алгоритму обробки природної мови використаний модуль spaCy та реалізація англійської NLP- моделі.

Для реалізації алгоритму вилучення напівструктурованих виразів використана бібліотека textacy.

Текст написаний на підставі особистого досвіду авторів, а також з використанням таких джерел [1, 2, 3]

ЛІТЕРАТУРА

1. Закон України «Про освіту» від 05.09.2017 № 2145-VIII. URL: http://search.ligazakon.ua/l_doc2.nsf/link1/t172145.html
2. NLP – это весело! Обработка естественного языка на Python. URL: <https://proglab.io/p/fun-nlp>
3. Основы Natural Language Processing для текста. URL: <https://habr.com/ru/company/Voximplant/blog/446738/>

Соколов К.О. Начальник Управління

Гудима О.П., к.т.н., с.н.с., Начальник відділу інформаційних ресурсів

Управління інформаційних технологій

Міністерства оборони України, м. Київ, Україна

Питання забезпечення кібербезпеки у всесвітньому інформаційному середовищі

Світова тенденція щодо зростання кількості користувачів сервісів соціальних інформаційних мереж (Інтернет) продовжує зберігати динаміку до зростання.

Це пов'язане з тим, що всесвітнє інформаційне середовище (Інтернет) має суттєві переваги над звичайними засобами і технологіями, а саме:

оперативність (розміщення і регулярне оновлення інформації не вимагають значного часу на підготовку матеріалів в електронному вигляді. Користувачі отримують її в режимі реального часу);

економічність (залучення невеликої кількості персоналу і матеріальних засобів для вирішення поставлених завдань);

скритність джерела впливу (як правило, акт агресії в глобальній мережі важко відрізнити від дії звичайних комп'ютерних хуліганів).

Інтернет надає широкі можливості для впливу на формування громадської думки, прийняття політичних, економічних і військових рішень, впливу на інформаційні ресурси опонента і поширення спеціально підготовленої інформації (дезінформації).

Підготувати та провести кібератаку з використанням всесвітнього інформаційного середовища (Інтернета) може досить широке коло осіб – від військових і розвідувальних структур іноземних держав до партизанських формувань, злочинців, промислових конкурентів, хакерів або просто озлоблених людей. Відстежити ж джерело досить складно.

Таким чином всесвітнє інформаційне середовище стало новим майданчиком щодо інформаційного впливу, що має ряд суттєвих переваг над класичними методами впливу.

Питанням інформаційної та кібернетичної безпеки в електронних засобах масової інформації (Інтернеті) в країнах світу приділяється суттєва увага.

На саміті НАТО в Варшаві (7-9.07.2016) на Кібер-конференції з інформаційного забезпечення НАТО (NIAS) (6.12.2016), на конференції "Кібернетична оборона" (Париж, 15.05.2018) та на засіданні Північноатлантичної ради (Брюссель 11-12.07.2018) було зосереджено увагу на важливості запобігання, виявлення та ліквідації деструктивного впливу (інформаційних загроз) у кіберпросторі (Інтернеті).

Відповідно до меморандуму 2017 р. у Гельсінкі (Фінляндія) за участю США, Франції, Німеччини, Швеції, Польщі, Фінляндії, Латвії, Литви створено Європейський центр з протидії гібридним загрозам. Який є міждержавним, європейським Центром боротьби з гібридними загрозами – кібератаками, пропагандою та дезінформацією. Передбачається, що Центр формуватиме мережу експертів для країн-учасниць та буде тісно співпрацювати з Євросоюзом і НАТО.

Враховуючи світові тенденції та на виконання завдань визначених керівними документами держави набувають актуальності питання моніторингу кіберпростору (Інтернету) та виявлення деструктивних кібервпливів (на етапах планування, підготовки та безпосередньо проведення інформаційних дій). Для реалізації вище зазначеного необхідна розробка відповідних моделей і методів.

При розробці методів (моделей) моніторингу соціальних мереж (Інтернету) можуть використовуватися нейронні мережі, які застосовуються до різних типів даних, але мають ряд недоліків, а саме: складність змістовної інтерпретації нейронних мереж та недетермінованість. Моделі на базі нейронних мереж не дозволяють однозначно та прозоро визначити внесок кожного показника у кінцевий результат.

Сьогодні проблема пошуку інформації у великих масивах є дуже актуальною. Ця проблема ускладнюється ще й тим, що сучасні пошукові механізми здійснюють пошук інформації без урахування семантики слів, що входять до запиту, а також контексту, в якому вони використовуються.

Для фільтрації контенту одним з сучасних рішень, щодо забезпечення ефективного моніторингу за різними станами інформаційних процесів, та їх впливу на суб'єктну діяльність, є онтологічне моделювання. Теорія графів дозволяє в свою чергу провести аналіз джерел інформації.

Використання онтологічного моделювання дозволяє здійснювати семантичний пошук, а з врахуванням алгоритму та підходів країн НАТО, що стосується підготовки до інформаційних операцій, дозволить сформувати науково-методичний апарат (удосконалити існуючий) по виявленню деструктивної інформації (кібервпливів) в кіберпросторі (Інтернеті) з урахуванням специфіки роботи органів військового управління Збройних Сил України.

Petrivskiy V., PhD student,
Shevchenko V., DSc, prof.,
Bychkov O., DSc, associated prof., Head of department
 Program systems and technologies department,
 Taras Shevchenko National University of Kyiv

Promising use cases 5G technologies in Internet of Things and Internet of Everything

The fifth generation of mobile and wireless networks is the next phase of mobile telecommunications standards [1]. 5G networks will include support of a large number of connected devices and flexible air interfaces and different interworking technologies that are energy efficient and will possess always online capabilities [1]. A major benefit of 5G is that it significantly lowers latency – the time lag it takes for two or more devices to talk to one another. The latency range for 4G is about 30-60 milliseconds. But with levels as low as 1ms, 5G offers “near-zero” latency. 5G offers major improvements in throughput as well. Standards adopted by the International Telecommunication Union (ITU) specify total download capacity for a single 5G mobile cell of at least 20Gbps [2]. By contrast, the peak data rate for current LTE cells is a relatively paltry 1Gbps. According to this standard there are many advantages and use cases of using 5G in Internet of Things and Internet of Everything, such as [3]:

- Massive IoT (also known as massive machine-type communications, or mMTC) applications and systems that can establish ongoing two-way communications with vast numbers of geographically dispersed IoT devices.
- Critical IoT (also called cMTC) applications to monitor and control assets in real time where latency requirements are very low and reliability needs high. For example, remote medical examination and surgery require very low latency and a high level of control so the surgeon gets tactile feedback that's indistinguishable from – or even better than – manual techniques.
- Media on demand applications will support consumer use of media content anytime, anywhere. Media delivery can be optimized for bandwidth, latency and cost by deploying a content distribution network close to the edge, thus allowing consumers to experience VR and AR with ease.
- Smart city networks that include remote monitoring of city infrastructure, real-time traffic information and public safety alerts for improved emergency response times.
- Emergency communications to help with search and rescue and identify and fix catastrophic problems involving machinery – even if parts of a network have been damaged.
- Smart agriculture that combines data from people, places and things with (for example) environmental data to automate systems that support worker health and productivity, along with regulatory compliance and the health/status of crops and equipment.
- Factory cell automation, including assembly line devices that communicate with control units with high reliability and low latency to support critical functions. Industrial facilities are expected to be among the biggest users of private 5G networks that are more secure and reliable.
- Smart venues and stadium networks that offer audiences a blend of physical and virtual experiences during concerts and sporting events, and allow crowd sourcing.

According to these use cases using communication technologies of the fifth generation will highly improve using Internet of Things and Internet of Everything devices and software.

REFERENCES

1. Mavromoustakis C. Internet of things (IoT) in 5G mobile technologies / C. Mavromoustakis, G. Mastorakis, J. Batalla., 2016. – 498 c.
2. Elkhord M. Networks of the future: Architectures, Technologies, and Implementations / M. Elkhord, Q. Hasssan, S. Shahrestani., 2018. – 489 c.
3. The impact of 5G on IoT Applications and Technology [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://www.sixgill.com/2018/08/09/5g-changes-ioe-governance-game/>.

Ващук Ірина, студент магістратури,
кафедра Інженерії програмного забезпечення
Факультет кібербезпеки, комп'ютерної і програмної інженерії
Національний авіаційний університет, м. Київ, Україна
Поперешняк С.В., к.ф.-м.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Автоматизація обліку та моніторингу робочого часу на підприємстві згідно чинного законодавства

Один з фундаментальних процесів управління персоналом на підприємстві є облік та моніторинг робочого часу. Документообіг - невід'ємна частина цих процесів, що може у декілька разів покращити ефективність роботи загалом, аби ж навпаки – уповільнити її чи використовувати надмірну кількість ресурсів.

За останнє десятиліття складність завдання управління кадрами в великих і особливо багатопрофільних підприємствах і організаціях значно зросла. В першу чергу це пов'язано з тим, що кадри є найважливішим ресурсом для забезпечення виробничо-господарської діяльності підприємства. Ефективність УП забезпечує ефективність діяльності самого підприємства і тому якісне управління персоналом є важливою конкурентною перевагою.

Українським законодавством визначено перелік первинних документів, що обов'язково повинні вестися підприємством. До них належать «Наказ про прийняття на роботу», «Наказ про надання відпустки», «Розрахунково-платіжні відомості» та інші. На основі таких первинних документів підприємство може формувати аналітичні звіти – вторинні документи, здійснювати виплату заробітної плати та сплату податків.

Формування як первинних, так і вторинних документів – трудомісткий процес, що потребує точності розрахунків та обробки великого обсягу даних, врахування всіх деталей відпрацьованого часу кожним співробітником.

Система управління персоналом, яка склалася на провідних вітчизняних підприємствах під впливом запровадження прогресивних зарубіжних технологій управління персоналом та викристалізації власного досвіду, включає сьогодні, функціональні підсистеми представлені на рис.1.



Рис. 1. – Функціональні підсистеми відділу персоналу

Для автоматизації процесів обліку та моніторингу робочого часу пропонується методика поступового перетворення накопичуваних даних у первинні та вторинні документи інформаційною системою із первинним внесенням даних кожним працівником чи відповідальними особами. Суть методики полягає у проходженні наступних етапів:

1. Внесення до інформаційної системи первинних даних кожного співробітника.
2. Створення шаблонів робочих графіків, за якими працює персонал.
3. Формування робочих графіків на основі створених шаблонів.
4. Присвоєння кожному співробітнику таблицю робочого часу, первинна інформація якого – заданий робочий графік, де всі не змінні робочі дні приймаються як відпрацьовані.
5. Внесення фактичних змін до таблиця кожним співробітником, або відповідальною особою: відсутність із зазначенням причини (хвороба, відпустка та ін.) та додаткових даних (вид відпустки), перевиконання / недовиконання норми робочого часу.
6. Щомісячне автоматичне формування кадрових документів системою.
7. Перевірка кадрових документів HR-менеджером та їх затвердження.
8. Автоматичне формування документів нарахувань заробітної плати та інших нарахувань на основі попередньо сформованих даних та інших первинних даних системи (формула розрахунку кожного виду нарахування).
9. Автоматичне обчислення податків на основі документів нарахувань.
10. Перевірка документів нарахувань та затвердження їх бухгалтером.
11. Автоматичне формування документів виплат, затвердження їх бухгалтером та фактичне проведення банківських операцій.
12. Автоматичне створення первинної та вторинної документації, створення аналітичних звітів, зведених таблиць та іншої інформації, необхідної підприємству.

Як ми бачимо, при реалізації даної методики інформаційною системою більшість дій виконуються автоматично. Завдання персоналу – початкове налаштування системи, забезпечення її всіма необхідними даними. За цією схемою значно зменшується час персоналу відділу кадрів на обробку інформації, формування документації, обчислення та перевірку даних. Оскільки у систему потрапляє завжди фактична й актуальна інформація про дотримання графіку кожним співробітником, зменшується ймовірність помилок та необхідність редагування затверджених документів через людську помилку. Таке поступове автоматичне перетворення даних пришвидшує роботу відділу управління персоналом та покращує його продуктивність.

В цілому розроблений програмний продукт дозволяє полегшити роботу осіб, відповідальних за процес управління персоналом на підприємстві, зменшити вірогідність виникнення помилок при обрахунках, автоматизувати процес документообігу підприємства та налаштувати гнучкий процес нарахування заробітної плати співробітникам. Програма дозволяє автоматично заповнювати документи на основі даних, введених у довідники та друкувати документи у форматі PDF. Система має інтуїтивно зрозумілий, простий у використанні інтерфейс та мінімальні вимоги до обладнання, оскільки для використання системи потрібен лише веб-браузер та доступ до мережі Інтернет.

Програмний засіб впроваджено як модуль до ERP-системи CS ERP.

ЛІТЕРАТУРА

1. Конституція України: [зі змін. та допов., внесеними Законом України від 1 лют. 2011 р. № 2952-VI]. – Харків : Фактор, 2011. – 118 с.
2. Господарський кодекс України від 16 січня 2003 року.
3. Cockburn A.: Writing Effective Use Cases, MA.: Addison-Wesley, 2001.
4. Meyer J.D. Руководство Microsoft по проектированию архитектуры приложений / J.D.Meier, David Hill, Alex Homer, Jason Taylor та ін.: [2-е изд.].– Microsoft. Patterns & practices, 2009. – 529 с.

Соколенко Поліна, студент магістратури,
кафедра Інженерії програмного забезпечення
Факультет кібербезпеки, комп'ютерної і програмної інженерії
Національний авіаційний університет, м. Київ, Україна
Поперешняк С.В., к.ф.-м.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Система обробки даних паркування автомобілів, як частина Smart City

На сьогоднішній день інформаційні технології активно впроваджуються у діяльність міст по всьому світу. Застосування технологій «розумного міста» використовуються для підвищення якості, продуктивності і інтерактивності міських служб, зниження витрат і споживання ресурсів, поліпшення зв'язку між міськими жителями і державою.

Розглянемо процес контролю паркування автомобілів у рамках «розумного міста». Дана тема буде актуальна як для міст України, так і за її межами, адже у багатьох містах збільшується кількість автомобілів і як наслідок виникає потреба у контролі процесів паркування, відслідковування порушень на дорогах, розміщення автомобілів на штраф-майданчиках тощо.

Smart Parking буде поширюватися і впроваджуватися в «розумні міста» в різних країнах світу. І тому є актуальним питання зручної організації управління даними та налаштуваннями процесів. А саме процесів взаємодії обладнання (яке є невід'ємною частиною «розумного міста»), паркувальних служб, контролюючих органів та водіїв.

Одною з головних особливостей «розумних міст» є використання обладнання, що допомагає отримати дані (датчики, камери тощо). Опираючись на ці дані можна мінімізувати людське втручання в процес розміщення авто, проте для кращої взаємодії виникає потреба у налаштуванні динамічних показників. Наприклад, ціна на паркування може змінюватися з плином часу або в залежності від території. Тому виникає потреба у системі управління такими динамічними показниками.

Smart Parking Manager – програмний засіб, що виконує функцію організації роботи з даними процесу паркування. Мета автоматизації – спростити процес роботи з обладнанням та даними, що беруть участь у процесі паркування транспортних засобів та реалізувати оптимальний спосіб організації даних.

Створюваний програмний засіб буде мати робочу назву "Smart Parking Manager" (SPM). Використання системи передбачається на пристроях з такими операційними системами, як Windows, Linux, MAC OS. Обов'язковою умовою, для використання даного програмного продукту є наявність браузера та доступ до мережі Інтернет, оскільки створюваний продукт є веб-застосунком.

Область застосування: паркувальні майданчики, штрафмайданчики, контролюючі органи, органи міського управління.

Впровадження інформаційних технологій в діяльність паркувальних майданчиків є необхідною умовою для покращення роботи служб контролю, оскільки оперативність, надійність і висока швидкість обробки та передачі інформації впливає на ефективність робіт.

Smart Parking Manager – це професійно-орієнтований комплекс, що включає в себе технічне, інформаційне і програмне забезпечення (рис. 1).



Рис. 1 – Складові системи

Технічне забезпечення

Головною вимогою до технічного забезпечення є наявність пристрою з підключенням до мережі Інтернет та встановленим сучасним браузером.

Рекомендована конфігурація:

Оперативна пам'ять: 512 MB і більше;

Об'єм жорсткого диска: 350 MB і більше;

Операційна система: Windows 7, 8, 10; Mac OS X 10.6 або пізніша версія; Ubuntu.

Вимоги до сервера

Оскільки створюваний продукт є веб-застосунком, важливо зазначити характеристики серверної сторони. Дане середовище потенційно буде виконувати операції над великим обсягом інформації, який з часом буде збільшуватися. Тому необхідний потужний сервер з можливістю розширення і збільшення об'єму оперативної пам'яті та засобів зберігання інформації на жорстких дисках.

Вимоги до локальної мережі.

Для досягнення оптимальних споживчих якостей програмних продуктів, які працюють на основі архітектури «клієнт-сервер» необхідна наявність високошвидкісної транспортної мережі. Рекомендується застосування мережі стандарту Ethernet або WiFi, що забезпечує оптимальну пропускну здатність мінімум 10 Мбіт / сек.

Архітектура програмного засобу. В даній роботі для створюваного програмного засобу було визначено компоненти трьох рівнів, кожен з яких належить відповідному шару.

Компоненти першого рівня представляють собою видиму для користувача частину – «веб-додаток», доступ до якого у користувача може отримати з будь-якого браузера. Користувач безпосередньо взаємодіє з цим рівнем, який в свою чергу відправляє запити до шару бізнес-логіки.

Компоненти другого рівня – бізнес-логіка («завдання» або «процеси» – спеціалізовані програми обробки) здійснюють ініційовану користувачем обробку даних, і повертають оброблені дані в БД, повідомляючи користувачеві через екран клієнтського додатка про завершення запитаної обробки.

Компонент третього рівня (прихована від користувача частина системи) – сервер бази даних, що здійснює зберігання даних. Користувач через клієнтську програму ініціює процедуру запиту на вибірку, введення, зміну або видалення даних у базі даних (БД).

Подальше застосування об'єктно-орієнтованого підходу до програмування цих компонентів підвищить можливості повторного використання, тестування та гнучкість розроблюваного засобу.

В роботі програмна розробка побудована на основі трирівневої архітектури, що представлена на рис. 2.

Рівень клієнта — це інтерфейсний (зазвичай графічний) компонент, який представляє перший рівень, власне застосунок для кінцевого користувача. Перший рівень не повинен мати прямих зв'язків з базою даних (за вимогами безпеки), не повинен бути навантаженим основною бізнес-логікою (за вимогами масштабованості) і зберігати стан програми (за вимогами надійності). На перший рівень буде винесена найпростіша бізнес-логіка: інтерфейс авторизації, перевірка значень, що вводяться, на допустимість і відповідність формату, нескладні операції (сортування, групування, підрахунок значень) з даними, вже завантаженими на термінал.

Сервер застосунків розташовується на другому рівні. На другому рівні зосереджена основна частина бізнес-логіки. Поза ним залишаються фрагменти, що експортуються на термінали, а також розміщені в третьому рівні збережені процедури і тригери.

Сервер бази даних забезпечує зберігання даних і виноситься на третій рівень. Це стандартна реляційна СУБД. Якщо третій рівень представляє собою базу даних разом з збереженими процедурами, тригерами і схемою, яка описує застосунок в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що зв'язує клієнтські компоненти з прикладною логікою бази даних.

Рівень клієнта — це інтерфейсний (зазвичай графічний) компонент, який представляє перший рівень, власне застосунок для кінцевого користувача. Перший рівень не повинен мати прямих зв'язків з базою даних (за вимогами безпеки), не повинен бути

навантаженим основною бізнес-логікою (за вимогами масштабованості) і зберігати стан програми (за вимогами надійності). На перший рівень буде винесена найпростіша бізнес-логіка: інтерфейс авторизації, перевірка значень, що вводяться, на допустимість і відповідність формату, нескладні операції (сортування, групування, підрахунок значень) з даними, вже завантаженими на термінал.

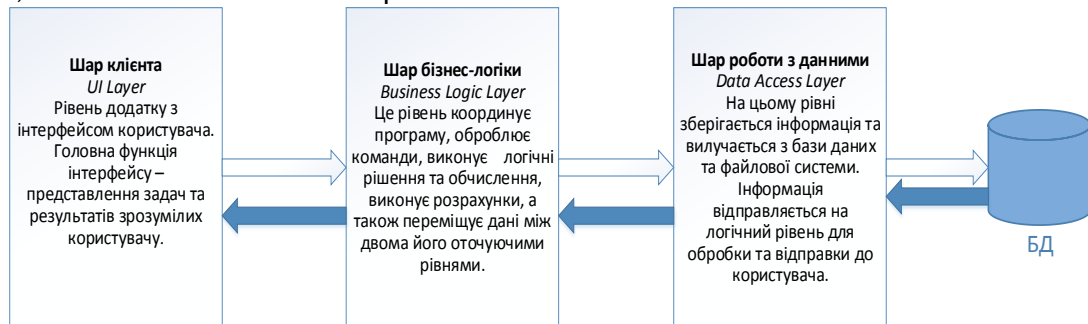


Рис. 2 – Триврівнева архітектура створюваного програмного засобу

Сервер застосунків розташовується на другому рівні. На другому рівні зосереджена основна частина бізнес-логіки. Поза ним залишаються фрагменти, що експортуються на термінали, а також розміщені в третьому рівні збережені процедури і тригери.

Сервер бази даних забезпечує зберігання даних і виноситься на третій рівень. Це стандартна реляційна СУБД. Якщо третій рівень представляє собою базу даних разом з збереженими процедурами, тригерами і схемою, яка описує застосунок в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що зв'язує клієнтські компоненти з прикладною логікою бази даних.

Інформаційне забезпечення SPM

В основі SPM лежить використання БД, що містить інформацію, необхідну для виконання службових обов'язків на даному робочому місці.

SPM буде забезпечувати зберігання даних про тарифи, штрафи, штрафні роботи, зони паркування, працівників, що є користувачами системи.

Для роботи з базою даних необхідно реалізовані такі функції:

- виведення вмісту бази даних;
- додавання записів у базу даних;
- редагування записів в базі даних;
- видалення записів з бази даних;
- пошук записів в базі даних.

БД повинна бути спроектована таким чином, щоб задовольняти вимогам щодо реакції системи на запити.

Доступ до даних має бути наданий лише авторизованим користувачам. Для забезпечення цілісності даних повинні використовуватися вбудовані механізми СУБД.

БД повинна бути спроектована таким чином, щоб задовольняти вимогам щодо реакції системи на запити. Для оператора задовільним є 1-3 секунди.

Інформаційна система для користувача базується на інформаційній моделі процесу взаємодії обладнання, працівників паркувальних служб, транспортних засобів та водіїв (рис. 3). Інформаційне забезпечення повинно вирішувати для його професійної діяльності наступні задачі – давати зручний доступ до даних з різних компонентів загальної системи.

Основна функція інформаційного забезпечення Smart Parking – забезпечити адміністрування процесів паркування транспортних засобів та розміщення їх на штраф-майданчиках. Таким чином, під час системного аналізу ПО слід виділити два напрямки дослідження бізнес-процесів: це аналіз даних з моніторингового обладнання та налаштування процесів.

Дані до бази можуть потрапляти наступними способами:

1. З обладнання що встановлене на паркувальних майданчиках та у місті. Технологія Smart City передбачає наявність датчиків встановлених по всьому місту для збору даних.

2. Введені користувачами з мобільних пристроїв на місці паркування або оформлення правопорушення.

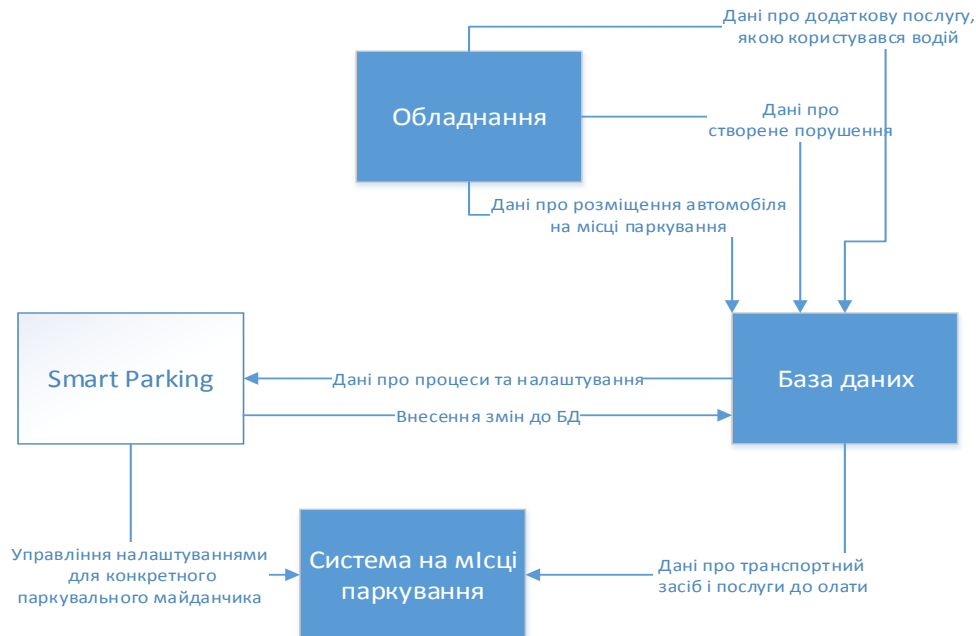


Рис. 3 – Місце Smart Parking в схемі процесу паркування як частини Smart City

Оптимізація обробки та аналізу даних процесу розміщення транспортних засобів на паркувальних майданчиках є необхідністю для великих міст. Розміщенням може стати система, яка буде обробляти дані з усього обладнання, що встановлене у межах міста, аналізувати ці дані, а також виконувати налаштування для паркувальних служб. Продукт дозволить оптимізувати процес роботи та виявити слабкі місця, для майбутнього покращення.

Висновки. На сьогоднішній день розвиваються системи, що спрощують процес паркування для користувачів(водіїв транспортних засобів), проте виникла необхідність у автоматизації адміністративної частини цього процесу. Це покращує роботу як операторів, так і органів міської влади, які зможуть контролювати та налаштовувати процес розміщення транспортних засобів на спеціально відведених місцях. Також це стане вагомим внеском у боротьбі з шахрайством пов'язаним з несплатою за паркування чи штрафів за правопорушення.

Створена система вирішує ці проблеми, при цьому варто зазначити, що система призначена не для певного міста, а є універсальним рішенням, що може бути адаптованим під будь-яке місто.

Технології Smart City вже активно впроваджується в багатьох містах по всьому світу, і дана система зможе знайти свою нішу, та покращити показники ефективності для міст що їх імплементують.

ЛІТЕРАТУРА

1. Rajendra Joshi. «Smart Cities: Breaking the Poverty Barrier» - M: Notion Press, 2019.
2. Системи автосервісу [Електронний ресурс] – Режим доступу: [https:// smartparking systems.com/en/smart-cities-cities-of-tomorrow/](https://smartparkingsystems.com/en/smart-cities-cities-of-tomorrow/)
3. SMART PARKING [Електронний ресурс] – Режим доступу: <https://www.happiestminds.com/whitepapers/smart-parking.pdf>

Поперешняк С.В., к.ф.-м.н., доцент кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Деякі аспекти тестування псевдовипадкової послідовності малої довжини

У сучасному світі велике значення мають генератори випадкових і псевдовипадкових послідовностей біт. Вони широко застосовуються в різних задачах: моделювання, вибіркові методи, чисельний аналіз, програмування, криптографія. Випадкові і псевдовипадкові послідовності (ПВП) мають величезну роль для криптографії – від їх якості залежить секретність інформації. Тому завдання створення хороших генераторів і ефективних методів їх оцінки становить великий інтерес [1].

Аналіз ефективності генераторів псевдовипадкових послідовностей є нагальною проблемою кібербезпеки за умов використання більш досконалих методів шифрування та захисту інформації. Наявні способи показують низьку гнучкість та універсальність у засобах знаходження прихованих шаблонів у даних. Для вирішення цієї проблеми запропоновано використовувати алгоритми на основі багатовимірних статистик. Дані алгоритми поєднують усі переваги статистичних методів та є єдиною альтернативою для аналізу послідовностей короткої та середньої довжини.

Проблеми малих і великих вибірок відносяться до основних проблем, що виникають при практичному застосуванні методів аналізу даних. Будемо використовувати наступну класифікацію вибірок за чисельністю [2], виходячи з вимог представлених в програмі критеріїв:

- дуже малі вибірки - від 5 до 12,
- малі вибірки - від 13 до 40,
- вибірки середньої чисельності - від 41 до 100,
- великі вибірки - від 101 і вище.

Мінімальну чисельність вибірки лімітує не так алгоритм обчислення критерію, скільки розподіл його статистики. Так, для ряду алгоритмів при занадто малих чисельностях вибірки нормальна апроксимація розподілу статистики критерію буде під питанням.

Для перевірки якості генераторів застосовуються різні тести. У світі немає будь-якого єдиного «офіційного» набору критеріїв, який би оцінював, наскільки дані випадкові послідовності біт можна використати саме для конкретної області застосування. Існують різні тести, які оцінюють, наскільки досліджувана послідовність біт «схожа» або «не схожа» на дійсно випадкову. Методи оцінки якості генераторів випадкових та псевдовипадкових послідовностей можна розділити на дві групи[1]:

1) Графічні тести. Властивості послідовностей відображаються у вигляді графічних залежностей, з вигляду яких роблять висновки про властивості досліджуваної послідовності.

2) Статистичні тести. Статистичні властивості послідовностей визначаються числовими характеристиками. На основі оціночних критеріїв робляться висновки про ступінь близькості властивостей аналізованої і істинно випадкової послідовностей.

Для тестування послідовностей на випадковість існує велика кількість алгоритмів, а для зручності перевірки послідовностей вже реалізовані програмні продукти, що містять в собі деякі набори тестів. Серед них найбільш поширені тести NIST STS, DIEHARD, CRYPT-X, тести Д. Кнута тощо.

Актуальність Незважаючи на чималу кількість існуючих реалізацій статистичних тестів ПВП [3], даний напрямок постійно розвивається. Аналіз існуючих пакетів [1-2] для статистичного тестування призводить до висновку, що багато тестів можуть успішно використовуватися для дослідження ПВП на випадковість за умови довжини послідовності понад 10^6 біт. Разом з тим, особливості прикладних задач показують, що класичні математичні моделі статистичного тестування не цілком адекватно відображає потреби в дослідженні послідовностей довжиною до 100 біт на випадковість [4-5].

Мета. Розглянути сумісні розподіли 2-ланцюжків і 3-ланцюжків фіксованого виду випадкової (0, 1)-послідовності, які дозволяють проводити статистичний аналіз локальних ділянок цієї послідовності.

Наукова новизна. В роботі запропоновано критерій для перевірки на випадковість (0, 1)-послідовностей невеликої довжини (до 100 біт).

Основний зміст. Опишемо відповідну математичну модель. Дано

$$\gamma_1, \gamma_2, \dots, \gamma_n, \quad (1)$$

де $\gamma_i = 0, 1$, $i = 1, 2, \dots, n$, $n > 0$.

Підпослідовність $\gamma_j, \gamma_{j+1}, \dots, \gamma_{j+s-1}$, послідовності (1) називається s-ланцюжком, $j = 1, 2, \dots, n - s + 1$, $s = 1, 2, \dots, n$.

Позначимо $\eta_{t_1, t_2, \dots, t_s}$ число s-ланцюжків в послідовності (1), які співпадають з t_1, t_2, \dots, t_s , де $t_i = 0, 1$, $i = 1, 2, \dots, s$.

Умова (У): послідовність (1) складається з n , $n > 0$, незалежних однаково розподілених випадкових величин; $P \gamma_i = 1 = p$, $P \gamma_i = 0 = q$, $p + q = 1$, $i = 1, 2, \dots, n$.

Твердження. Нехай виконуються умова (У), k_1, k_2, k_3, t, t_1 – цілі числа такі, що $k_1 \geq 0, k_2 \geq 0, n \geq k_1, k_3 \geq 0, t, t_1 \in 0, 1$. Тоді

$$P \eta_{t_1 t_1^*} = k_1, \eta_{t_1 t} t_1^* = k_2 = \frac{n-k_1}{m_1=k_1} p^{m_1} q^{m_0} C_{k_1}^{k_2} C_{m_1-k_1}^{k_2} C_{m_1}^{k_1}$$

де $m_0 = n - m_1$, символ \sum позначає сумування по усім невід'ємним числам δ_0 та δ_1 таким, що $\delta_0 + \delta_1 = 2k_1 - k_2$, $t^* \stackrel{\text{def}}{=} 1 - t$.

Експеримент.

В результате применения данной методики для тестирования псевдослучайных последовательностей построены таблицы, с помощью которых можно получить вероятность распределения нулей и единиц в заданной последовательности. Как показывает практика использование готовых таблиц для анализа последовательности на случайность позволяет максимально быстро получить ответ, в отличии от классического метода тестирования.

В результаті застосування даної методики для тестування псевдовипадкових послідовностей побудована таблиця, за допомогою якої можна отримати ймовірність розподілу нулів і одиниць в заданій послідовності. Як показує практика використання готових таблиць для аналізу послідовності на випадковість дозволяє максимально швидко отримати відповідь, на відміну від класичного методу тестування.

Розглянемо приклад таблиці для бітової послідовності малої довжини. Наприклад, нехай довжини бітової послідовності n , $n=20$.

Проілюструємо (табл. 1, рис. 1) використання твердження для малої вибірки n , $n = 20$, і деяких значень k_1 та k_2 .

В таблиці 1 в першому стовпчику наведено всі можливі варіанти значень k_1 і k_2 , для яких ймовірність $P \eta_{t_1 t_1^*} = k_1, \eta_{t_1 t} t_1^* = k_2 \geq 0,001$. В другому стовпчику таблиці 1 подано ймовірності (в неспадному порядку) $P \eta_{t_1 t_1^*} = k_1, \eta_{t_1 t} t_1^* = k_2$ для пар чисел (k_1, k_2) , які вказані в першому стовпчику.

Таблиця 1. Таблиця ймовірностей зазначеної події в залежності від k_1 та k_2

(k_1, k_2)	P	(k_1, k_2)	P	(k_1, k_2)	P
(1, 1)	0,0011	(2, 2)	0,0111	(5, 1)	0,0546
(8, 2)	0,0021	(7, 3)	0,0121	(6, 3)	0,0573
(8, 1)	0,0022	(7, 1)	0,0134	(6, 2)	0,0716
(6, 5)	0,0026	(6, 4)	0,0195	(4, 3)	0,0927
(7, 0)	0,0029	(7, 2)	0,02	(5, 3)	0,1091
(3, 0)	0,0029	(4, 4)	0,0232	(4, 2)	0,1113
(7, 4)	0,003	(3, 1)	0,0245	(5, 2)	0,1227
(5, 5)	0,0042	(3, 3)	0,0303		
(4, 0)	0,0059	(6, 1)	0,0368		
(6, 0)	0,0061	(5, 4)	0,0382		
(2, 1)	0,0074	(4, 1)	0,0472		
(5, 0)	0,0076	(3, 2)	0,0531		

На рис. 1 представлено бульбашкову діаграму, в якій перший параметр (горизонтальна вісь) – значення k_1 , другий (вертикальна вісь) – значення k_2 , третій (розмір бульбашки) – ймовірність події $\eta_{t_1 t_1^* = k_1, \eta_{t_1 t_1^* = k_2}}$, яка подана в відсотках.

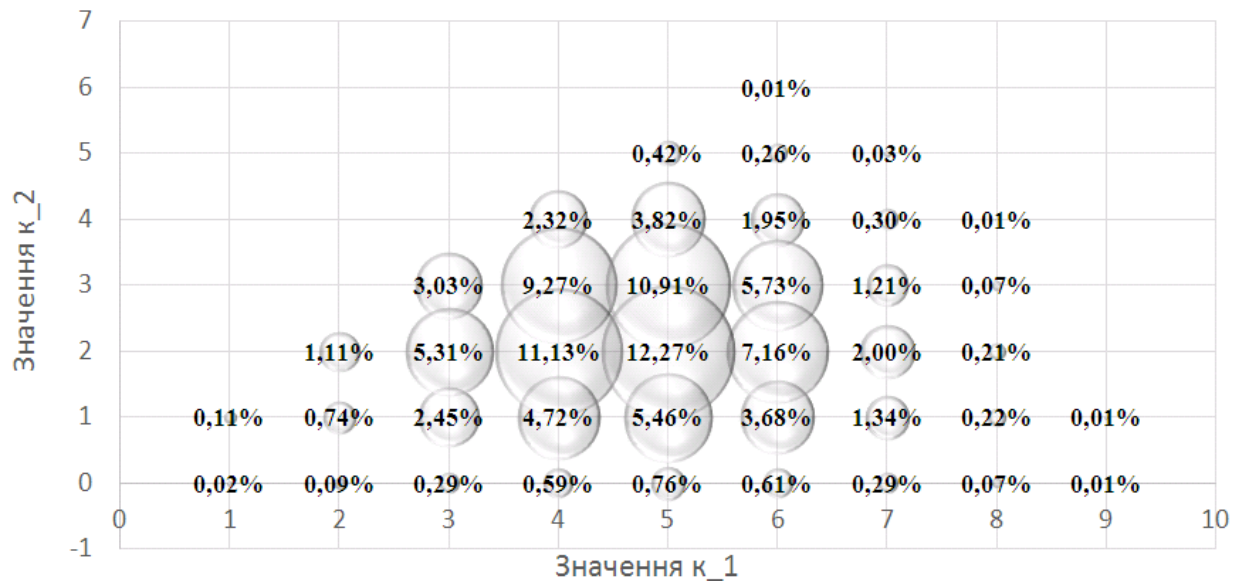


Рис.1. Розподіл ймовірностей зазначеної події в залежності від k_1 та k_2 .

Висновки. Тестування генераторів випадкових та псевдовипадкових послідовностей, які використовуються в криптографічних додатках, є актуальним завданням як у практичному, так і в теоретичному плані. Незважаючи на значні напрацювання в цій галузі, розробники проте відчувають потребу в зручному «універсальному» програмному засобі, здатному надати прийнятну метрику, яка дозволить досить ясно досліджувати ступінь випадковості послідовностей, що породжуються генераторами випадкових (або псевдовипадкових) чисел, крім того, забезпечити розробників достатнім обсягом інформації для прийняття рішення щодо «якості» генератора.

Підхід до тестування з використанням багатомірних статистик дозволяє розраховувати на більш глибоке обґрунтування випадковості бітових послідовностей, які аналізуються. Ця область є перспективною для наукових досліджень. Таким чином, в роботі запропонована нова методика тестування ПВП і розглянуто декілька критеріїв для тестування бітової послідовності малої довжини, який в порівнянні з одновимірними статистиками дають більш точний результат.

В даній роботі наведено точні сумісні розподіли деяких статистик $(0, 1)$ -послідовностей довжини n , $1 < n < \infty$. Для бітової послідовності малої довжини n , $n = 20$ наведені таблиці, що містять числові значення відповідного розподілу. Ці таблиці, так само як і запропоновані їх графічні представлення, можуть бути використані для перевірки гіпотези випадковості розташування нулів і одиниць.

ЛІТЕРАТУРА

1. Григорьев А. Ю. Методы тестирования генераторов случайных и псевдослучайных последовательностей // Ученые записки УлГУ. Сер. Математика и информационные технологии. УлГУ. Электрон. журн. – 2017. – № 1. – с. 22-28.
2. Гайдышев И.П. Программное обеспечение анализа данных AtteStat. Руководство пользователя. Версия 13 – 2012. – 505 с.
3. V Masol, S Popereshnyak A theorem on the distribution of the rank of a sparse Boolean random matrix and some applications // Theory of Probability and Mathematical Statistics/ – 2008. – V. 76. – P. 103-116
4. S Popereshnyak The Technique for Testing Short Sequences as a Component of Cryptography on the Internet of Things [Електронной пєсурп] <http://ceur-ws.org/Vol-2516/paper11.pdf>
5. S Popereshnyak, GP Dimitrov The Testing of Pseudorandom Sequences using Multidimensional Statistics [Електронной пєсурп] <http://ceur-ws.org/Vol-2533/paper14.pdf>

Прудченко А.О. студент 3 курсу
 кафедра Інженерії програмного забезпечення
Старкова О.В., к.т.н.,
Герасименко К.В., к.т.н.,
 кафедра Мережесих та Інтернет технологій
 Факультет інформаційних технологій
 Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Алгоритм співставлення зображень за ключовими точками при масштабуванні і обертанні об'єктів

Комп'ютерний зір — надзвичайно широка область, яка включає в себе багато різнопланових задач, таких як сегментація, фільтрація, класифікація, реконструкція сцени, оцінка положення об'єкта, виявлення об'єктів, відеоспостереження та багато інших. Комп'ютерне бачення є важливою складовою розвитку штучного інтелекту та інтелектуальних інформаційних технологій. Коло застосувань цієї технології розширюється, адже потреба у системах штучного інтелекту зростає, а зір — це один з найінформативніших сенсорів, який може використовуватись в таких системах.

Порівняння методів заснованих на аналізі інтенсивності та виділення ознак

В даному процесі, одна частина зображень являє собою опорне зображення або еталон, а інші відповідні зображення називаються цільовими, або тими, що підлягають скануванню, пошуку об'єктів. Реєстрація зображення досягається шляхом співставлення цільових зображень, так щоб вони збігалися з еталонним зображенням. У методах, що працюють з інтенсивністю зображень, порівнюють зразки інтенсивності зображення на основі кореляції показників, а в методах оснований на виділенні ознак шукають відповідності між елементами зображення, такими як точки, лінії і контури. Методи на основі аналізу інтенсивності, зіставляють цілі зображення чи частини зображень. Якщо зіставляються частини зображень, центри відповідних частин зображення розглядаються як відповідні характерні точки. Методи на основі виділення ознак головним чином встановлюють відповідність між групою різних точок зображення. Знайшовши відповідність між цими групами точок на зображенні (образами), для зображення кінцевого зображення виконується геометрична трансформація над цільовим зображенням, перед співставленням з еталонним зображенням, яка показує відповідність між цими зображеннями точка за точкою.

Методи пошуку зображень за ключовими точками

Ключові (або особливі) точки — це точки, за якими можна класифікувати зображення, розпізнати його, якась особливість зображення, унікальність. Для різних методів це різні особливості — кутові точки, точки різкої зміни кольору, яскравості, тощо. Ключовою вважається така точка зображення об'єкта, яка з великою часткою ймовірності буде знайдена на іншому зображенні цього ж об'єкта. Для визначення ключових точок зображення використовується метод детектору. Детектор повинен забезпечувати інваріантність знаходження одних і тих же особливих точок щодо перетворень зображення. Ключова точка має задовольняти певним умовам:

- визначеність (distinctness): точка має виділятися на фоні сусідніх точок;
- стійкість (repeatability): зміна яскравості, контрастності або кольорової гами не повинна впливати на розміщення ключової точки на зображенні;
- стабільність (stability): наявність шумів на зображенні не повинна впливати на роботу детектору;
- інваріантність (invariance): особливі точки мають бути стійкими до трансформацій зображення (масштабування, обертання, зміни ракурсу то що);
- інтерпретованість (interpretability): ключові точки мають бути представлені у зручному для подальшої роботи форматі.

При пошуку зображень значний вплив на результати пошуку мають наступні фактори:

- масштаб: зображення мають різний масштаб, в результаті чого предмети, які людина сприймає як однакові, займають різну площу на різних зображеннях;
- розміщення на зображенні: об'єкт може знаходитися в різних місцях зображення;
- фон: предмет на зображенні може бути ніяк не виділено, і перебувати на фоні інших предметів;
- перешкоди: зображення може бути піддано різного роду деформаціям;
- трансформації: проекція, обертання і кут огляду.

Зображення є лише двовимірною проекцією тривимірного світу, тому поворот об'єкту і зміна кута огляду кардинальним чином впливають на його двовимірну проекцію – зображення. Один і той же об'єкт може давати абсолютно різні зображення, в залежності від кута зору або відстані до об'єкту. Саме тому кожен метод виявлення ключових точок повинен гарантувати інваріантність щодо будь-яких перетворень зображення. Для визначення відповідності ключових точок подібних зображень кожній особливій точці присвоюється опис, який буде однаковим на різних зображеннях. Для цього формується дескриптор – ідентифікатор ключової точки, який робить її унікальною щодо інших особливих точок. Тобто дескриптор – це опис точки, який визначає особливості її щодо оточення. Дескриптор визначається як числовий або бінарний вектор і дозволяє виділити особливу точку з множини точок на зображенні, що при порівнянні зображень необхідно для складання пар особливостей, які належать одному об'єкту. Для виконання даної роботи буде використано метод пошуку SIFT.

SIFT (Scale-invariant feature transform)

Алгоритм SIFT знаходить характеристики зображення, які інваріантні до масштабування зображення і обертання, та частково інваріантні до змін в освітленні і точки знімання 3D камери. Виявлені характеристики значно відрізняються одна від одної, що дозволяє з високою ймовірністю коректно знайти відповідність між однаковими характеристиками та різних зображеннях, забезпечуючи основу для розпізнавання об'єктів. Алгоритм включає такі основні етапи:

1. Знаходження екстремумів у просторі масштабів. На першому етапі відбувається пошук по всіх масштабах і розташуваннях на зображенні.
2. Локалізація ключових точок. Для кожного місцезнаходження точки-кандидата знаходиться детальна модель для визначення розташування і масштабу. Ключові точки вибираються на основі міри їх стійкості
3. Визначення орієнтації. На основі локальних напрямків градієнта кожній ключовій точці призначаються одна чи декілька орієнтацій. Всі подальші операції виконуються на даних зображення, які були перетворені по відношенню до призначеної орієнтації, масштабу і місцеположення кожної характеристики, забезпечуючи тим самим інваріантність цих перетворень.

Висновок

В результаті виконання роботи буде розроблено ПЗ, яке зможе визначати наявність заданого автомобіля на зображенні. Для цього система має навчитися на попередньо підготованих зображеннях автомобілів.

ЛІТЕРАТУРА

1. Т.О. Паромова, І.Я. Зеленьова, Н.В. Луценко, Є.В. Білик. Порівняльний аналіз методів визначення ключових точок при пошуку зображень за фрагментами. Наукові праці ДонНТУ. Запоріжжя, 2018. №1 (26). С. 2.
2. В.М. Дичко, С.В. Ляхевич. Оптична навігаційна система бпла: порівняльний аналіз алгоритмів знаходження ключових точок. Міжнародний науковий журнал. Київ, 2016. №6, т. 1. С. 13
3. Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints / D.G. Lowe // International Journal of Computer Vision. — 2004. — Вид. 60, № 2. — С. 109–110.

Мирошников С.О., студент магістратури
Карнацький Д.О., студент магістратури
Бичков О.С., д.т.н., доцент, завідувач кафедри
кафедра Програмних систем і технологій
Київський національний університет імені Тараса Шевченка, м.Київ, Україна

Інформаційно-аналітична система електронного голосування з використанням протоколу двох агентств Фудзіоки-Окамото-Охта

У роботі представлено реалізацію електронного голосування на базі протоколу двох агентств Фудзіоки-Окамото-Охта. Було проаналізовано та досліджено найбільш поширені алгоритми для забезпечення анонімного одноосібного голосування і використав при розробці той, що відповідає найбільшій кількості вимог. Слід зазначити, що алгоритм був удосконалений задля забезпечення роботи системи загалом при виході зі строю її окремих компонентів.

Розроблене програмне забезпечення складається з трьох компонентів, які можуть розміщуватись на різних серверах, що забезпечує загальну стійкість системи і дотримання принципів голосування - таємність, можливість голосування тільки тих осіб, які мають таке право, неможливість зміни голосу та ін. Було розроблено саме веб-застосунок, як найбільш доступний та простий для використання кінцевим користувачем.

Схема Фудзіоки-Окамото-Охта, розроблена в 1992 році, ґрунтується на протоколі двох агентств та криптографічному підпису наосліп. Несильно ускладнюючи протокол, ця схема частково вирішує проблему змови двох агентств. Для роботи протоколу необхідний заздалегідь вибраний спосіб маскуючого шифрування, під яким виборець надсилає реєстратору бюлетень. Сліпе (маскуюче) шифрування — особливий вид шифрування, що дозволяє упевнитися в тому, що документ справжній і підписаний авторизованим користувачем, але не дає дізнатися, які дані в ньому містяться.

Обраний протокол двох агентств Фудзіоки-Окамото-Охта складається з двох компонентів – реєстратора та лічильника, має всі властивості крім публікації списку осіб, що проголосували, але це можна вдосконалити

Розроблена система відповідає наступним вимогам:

- Голосування тільки користувачами, які мають на це право (пройшли верифікацію, ще не голосували)
- Виборець не може проголосувати більше одного разу
- Голосування відбувається анонімно
- Неможливо проголосувати за іншого виборця
- Неможлива таємна підміна голосу виборця
- Відмовостійкість компонентів програмного забезпечення - при виході з ладу одного з компонентів система продовжує своє функціонування

Електронне голосування через мережу Інтернет значно прискорює підрахунок голосів, враховує інтереси людей з обмеженими можливостями, підвищує активність виборців.

Реалізоване програмне забезпечення рекомендується до впровадження у навчальних закладах різного рівня акредитації для виборів в органи студентського самоврядування.

Література

1. Протоколы голосования – Учебная и научная деятельность Анисимова Владимира Викторовича – Лекция 17. Криптографические методы защиты информации [Електронний ресурс]. – Режим доступу до ресурсу: https://sites.google.com/site/anisimovkhv/learning/kripto/lecture/tema15/tema15_3.
2. What Is Democracy? – Elections [Електронний ресурс]. – Режим доступу до ресурсу: <https://usa.usembassy.de/etexts/gov/democracy-elections.htm>.