

УДК 338.4: 65.015

УПРАВЛІННЯ ПІДПРИЄМНИЦЬКОЮ ДІЯЛЬНІСТЮ В УМОВАХ ЗРОСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Наумік-Гладка К. Г. - д.е.н., доцент,

Птащенко О. В. - к.е.н., доцент,

ДВНЗ «Харківський національний економічний університет імені Семена Кузнеця»

У статті розглянуто основні тенденції розвитку інформаційних технологій. Розглянуто взаємодію розвитку та управління підприємницькою діяльністю за умов використання інформаційних технологій. Подано пропозиції щодо удосконалення управління підприємницькою діяльністю на основі використання інформаційних технологій.

Ключові слова: підприємницька діяльність, управління, ІТ технологія, ІТ проект, ІТ компанія.

Наумик-Гладкая Е. Г., Птащенко Е. В. Управление предпринимательской деятельностью в условиях роста информационных технологий

В статье рассмотрены основные тенденции развития информационных технологий. Также уделено внимание взаимодействию развития и управления предпринимательской деятельностью при использовании информационных технологий. Представлены предложения по совершенствованию управления предпринимательской деятельностью на основе использования информационных технологий.

Ключевые слова: предпринимательская деятельность, управление, ИТ технологии, ИТ проект, ИТ компания.

Naumik-Gladka K. G., Ptashchenko O. V. Management of entrepreneurship activity in conditions of information technology growth

The article describes the basic trends of development of information technologies. Also attention is paid to the interaction of development and management of business activities by using information technology. Suggestions to improve the management of business activities based on the use of information technology are presented.

Keywords: Entrepreneurship, Management, IT technology, IT project, IT company.

Постановка проблеми. Сучасною особливістю розвитку суспільства у XXI столітті є збільшення ролі інформаційної сфери в житті людини у повсякденному житті, а відповідно й інформаційної діяльності. Саме тому використання інформаційних технологій у роботі сучасного суб'єкта господарювання є актуальною та з кожним днем все більш значущою. Сфера бізнес-послуг в області інформаційних технологій (ІТ) в наш час має високі темпи зростання [6]. Зокрема, на підставі результатів рейтингу «Інформаційні технології для промисловості – 2013» [6], можна зазначити наступне: кількість ІТ-компаній має тенденцію до постійного зростання; зростає кількість вітчизняних і закордонних програмних продуктів для промисловості; постійний (часто стрибкоподібний, а не рівномірний) приріст виторгу за останні 3 роки ІТ-компаній підтверджує велику кількість ніш і можливостей на ринку ІТ-продуктів для промислових підприємств.

Ріст масштабів ІТ-бізнесу одночасно означає збільшення обсягів послуг консалтингових компаній, пов'язаних з оптимізацією бізнес-процесів саме в ІТ компаніях. Слід зазначити, що діагностика процесів керування компанією в сфері ІТ актуальна в наступних випадках:

при плануванні: об'єднання ІТ-підрозділів у рамках злиття/поглинання компаній; збільшення співробітників ІТ-компанії; розвитку керування ІТ-компанією (у цьому випадку результати обстеження можуть використовуватися не тільки для планування робіт, але й для обґрунтування відповідних витрат); впровадження нових корпоративних інформаційних систем, що впливають на діяльність ІТ-підрозділів усередині самої компанії, наприклад системи менеджменту якості (СМК); розвитку виробничих напрямків в ІТ-компанії.

при оцінюванні: поточного стану об'єкта керування, наприклад: при зміні ІТ-керівника (або особи, що курирує ІТ на підприємстві); кадрового потенціалу; ступеню відповідності корпоративних стандартів окремих ІТ-компаній у складі холдингових структур.

Аналіз останніх досліджень і публікацій. Сьогодні дослідженням питань управління у ІТ та розвитку ІТ іноваційного бізнесу присвячені роботи багатьох авторів, наприклад, Б.Гейтса [1], Д. Белла [13], П. Друкера [3], В. Иноземцева [5], Д. Каштанова [7], М. Мариничевої [8], В. Чугреева [12].

Програмний код, що створює ІТ компанія, є її системотворчим фактором і базовим елементом. Таким чином, у компанії, що займається розробкою програмного забезпечення (software company), саме співробітники, їх компетенції й особисті якості, визначають її життєздатність і ринкову ефективність. Афоризм «кадри вирішують усе» актуальний і дотепер.

Успіх ІТ-компанії на ринку також залежить і від маркетингової складової, яку забезпечують менеджери проектів (Project managers, PM) і відділ продажів (Sales managers). Саме їх знання та досвід стають об'єктом керування системи управління знаннями.

Термін «управління знаннями» має різне тлумачення. З погляду організації практичної діяльності найбільш глибоке, на наш погляд, є таке: «управління знаннями (knowledge management) – це процес створення умов для виявлення, збереження й ефективного використання знань й інформації в співтоваристві; стратегія, спрямована на надання вчасно потрібних знань тим членам співтовариства, яким ці знання необхідні для того, щоб підвищити ефективність діяльності» [8]. Термін «знання» також неоднозначний, у даній статті він використовується в наступному значенні: «знання - це закономірності предметної області (принципи, зв'язки, закони), отримані в результаті практичної діяльності й професійного досвіду, що дозволяють фахівцям ставити й вирішувати завдання в цій області» [10].

Погодимося з [12] що можливо виділити дві головні складові систем управління знаннями: технічне забезпечення (технічна система, яка забезпечує нагромадження знань, а також доступ до них) і корпоративну культуру (система, яка сприяє обміну знаннями, їх нагромадженню й використанню).

У процесі економічної діяльності ІТ-компанії створюються знання, у вигляді програмного коду й супровідної документації (технічне завдання, аудіо- і відеоматеріали для роботи співробітників, переписка із клієнтами, інструкції для експлуатації та ін.). Створене знання необхідно оформити й зберегти, це технічне завдання. При цьому виникає питання визначення, що саме зберігати, у якому виді, яке саме й кому передавати знання. Це вже завдання корпоративної культури.

Якою би гнучкої й зручної не була технічна система, вона не здатна компенсувати відсутність корпоративної культури обміну знаннями. Отже, якщо немає зацікавлених в обміні або передачі знань людей або їх мало, то технічна система самостійно без мотиваційної підтримки персоналу не впорається. Знання не буде використовуватися. Інша сторона цього питання в тому, що при наявності зацікавлених, високомотивованих людей впровадження системи управління знаннями стає можливим навіть без серйозних фінансових вкладень у технічну складову.

Основний принцип технічної організації сховища – це структурований порядок розміщення інформації й зведений документ, у якому міститься, що й де перебуває, а також наведена коротка анотація по змісту.

У свою чергу основний принцип корпоративної культури розуміється як сукупність соціальних механізмів і цінностей, що сприяють (або перешкоджають) обміну знаннями, їх збереженню, використанню й створенню нових [2].

Таким чином, культура проявляє себе у формі традицій і норм поведінки [2], культура керування знаннями також може й повинна проявляти себе у вигляді відповідних норм і традицій, орієнтованих на обмін знаннями. Ці норми й традиції потрібно створювати, розбудовувати й системно впроваджувати як організаційно-методичні заходи щодо обміну знаннями [9].

Постановка завдання. Метою даної статті є визначення особливостей управління підприємницькою діяльністю в умовах зростання інформаційних технологій на прикладі ІТ компаній на основі корпоративної культури.

Виклад основного матеріалу досліджень. В ІТ-компанії управління підприємницькою діяльністю спрямовано на створення кінцевого продукту – коду. Приймаючи до уваги значення корпоративної культури як складової управління компанією, слід наголосити, що корпоративна культура повинна допомагати насамперед у створенні якісного програмного коду.

Є два основні показники створюваного коду: швидкість його написання і якість (якість вихідного коду, а не дієвість кінцевої програми). Існують різні методики [11] для оцінки цих показників. При цьому швидкість розробки традиційно стає пріоритетом. Це обумовлене як маркетинговими міркуваннями (ринок і клієнти не готові довго чекати), так і економічними (вартість розробки програмної системи залежить від строків її розробки) та визначає сегменти на ринку ІТ продукції (див. рис.).

max	ВАРТІСТЬ		max
ШВИД-КІСТЬ	Молоді фахівці		Senior фахівці
	Демпінг		Фахівці, що мають «Know How»
min			

Рис. Сегменти фахівців на ринку ІТ продукції

Приймаючи до уваги, що клієнти платять за швидкість написання коду, то з позиції внутрішнього середовища компанії: чим швидше код буде написаний, тим менше потрібно буде платити програмістам і, відповідно, тим менше собівартість розробки програми. Якщо головна і єдина вимога до коду це його працез-

датність, то якість коду буде ігноруватися, а оплата роботи з його поліпшення не запланована. Це значить, що компанія заощаджує на внутрішній якості, яка непомітно (або малопомітно) для кінцевого користувача.

З позиції зовнішнього середовища компанії: чим швидше пишеться код, тим погодинна ставка оплати роботи програміста повинна бути вищою. Це пов'язано з тим, що виникає питання значного демпінгу на ринку програмних продуктів.

Ключовим критерієм в цій ситуації стає якість створеного коду. У зв'язку із цим виникає потреба в експертних створеного коду, які можливо здійснювати за допомогою таких додатків, як: Google Skipfish, Codepad, CodepadIDE, Write Code Online, Tinkerbin, JS Bin, JsFiddle, Cloudsk, Jsdo.It, Google Code Playground, Code Run, Compilr або за допомогою програмістів рівня Senior.

Такий підхід має свої недоліки, найбільше точно ситуація описується через концепцію технічного боргу (Technical Debt), придуману Уордом Каннингемом (Ward Cunningham) [7, 12]. Слушним є зауваження Д. Каштанова, що технічна заборгованість не може бути дорівнює нулю в принципі (вірніше, теоретично може, але на дуже короткий строк) [7].

Отже, сутність питання полягає в наступному: розробники коду можуть одержати додатковий бонус на початковому етапі у вигляді прискорення розробки за рахунок використання швидко-розроблювальних схем та кодів, але неоптимальних технічних рішень (алгоритмів, архітектури). Згодом у процесі доробки й модифікації коду за такі рішення доводиться розплачуватися додатковими праце- і часо- витратами. Погасити «борг» можна за допомогою рефакторинга [7]. Таким чином, у проектах, що вимагають швидкого виходу на ринок, іноді трапляється, що якість замінена швидкістю на початковому етапі розробок програмного продукту. Але після того, як ринкова ніша зайнята, то можна повернутися до проблеми неякісного коду й «сплатити борг». Очевидно, що використання такого підходу як базової стратегії у всіх випадках недоцільно. Неякісний код служить не тільки причиною помилок, які досить важко виявити, але й причиною складності подальшого супроводу проекту. Іноді, за таких умов, може виникнути така ситуація, коли простіше переписати програму заново, чим виправляти те, що є. Таким чином, початкова економія обертається істотними втратами в майбутньому періоді.

Є ще один вагомий фактор, з яким неминуче зіштовхнеться компанія, - кадрові проблеми та демотивація персоналу, що знову набирається. Розбиратися в чужому, неякісно написаному коді досвідченим фахівцям, що цінують свої знання й час, не цікаво. Якщо керівництво при цьому орієнтується на швидкість написання неякісного коду, які демонструють малодосвідчені програмісти, але які вже працюють у компанії, і очікує таку ж швидкість від знов прийнятого на роботу програміста при доробці програми, то природно, що це нереалістичні очікування. Цілком може виявитися так, що досвідчений і кваліфікований програміст, змушений працювати із чужим, неякісним кодом, буде демонструвати гірші результати, чим його колеги з такою ж й навіть меншою кваліфікацією, що працювали із цим кодом раніше. Хвилювання, викликане адаптацією, незвичність нового робочого місця, неминучі втрати часу на соціалізацію та вибудовування відносин з колегами, нестача уваги з боку керівника - усе це може позначитися на ефективності роботи фахівця.

У сукупності ці фактори можуть сформувати ідею про певну неефективність прийнятого працівника. У підсумку – розчарування програміста в роботі, а також невдоволення керівництва й, як наслідок, - звільнення. Якщо такий сценарій відтворюється регулярно, то фірма буде змушена постійно шукати нових співробітників, тобто фактично займатися підбором кадрів на ринку праці, що спричиняє додаткові тимчасові й фінансові втрати.

Виходячи з вищезазначеного, можливо дійти розуміння важливості якості коду, а також своєчасної й систематичної роботи з його поліпшення. Корпоративна культура управління знаннями, безумовно, повинна враховувати цей момент.

Існує ряд стійких до теперішнього часу заходів обміну знаннями, проведених у software IT-компаніях. Дані заходи є як технічними (вони націлені на поліпшення коду), так і культурно-навчальними (вони дозволяють обмінюватися знаннями). Перелічимо їх у табл. 1.

Таблиця 1 Заходи обміну знаннями у software IT-компаніях

Заход	Зміст	Переваги
Перегляд коду (code review)	регулярний перегляд коду колективне обговорення написаного коду	крім виявлення помилок і усунення неоптимальних розв'язків дозволяє обмінюватися досвідом, удалими вирішеннями й підходами, а також приводить код проекту до однаковості. Якщо проект великий й програмісти спеціалізуються на окремих його частинах, перегляд і обговорення чужого коду дозволяє краще зрозуміти логіку й особливості роботи всієї системи
поліпшення корпоративної культури	працівники повинні знати: один одного особисто, що очікується від них, і мати інструменти й навички для виконання потрібних завдань	Дозволяє забезпечити чітке узгодження індивідуальних цілей і цілей організації, її місії й завдання

Перегляд коду ще називають інспекцією або ревізією коду, тобто вибирається чийсь код і аналізується. На обговоренні присутні автор і інші програмісти, які виступають у ролі рецензентів [7].

Приблизно у 1985 році соціолог і консультант Пітер Дракер вимовив крилату фразу, що стала: культура з'їдає стратегію на сніданок (culture eats strategy for breakfast) [14], що означає, що якщо в компанії не впроваджена правильна культура, те неважливо, який спосіб відстеження помилок використовується, якість робіт буде «зідено» автоматично й заздалегідь. Керівник проекту є джерелом культури в безпосередньому середовищі проекту, тому ми відповідальні за те, що Пітер Друкер називає більш ефективним, ніж стратегія діяльності [4].

Таким чином, робота над культурою фірми, особливостями її поведінки при веденні справ, настільки ж важливі, як і сама робота в компанії.

При роботі з IT-проектами використання спеціалізованих методів управління проектами буде дуже корисним для безперервного прогресу, що приводить до більшої успішності в результаті. Планування й виконання IT-проектів може бути важким за різних обставин, що робить уміння успішно їх завершувати необхідним для будь-якого роботодавця.

Аналізуючи практику діяльності ІТ-компаній можливо зазначити, що провалу ІТ-проектів сприяють наступні фактори: відсутність професіоналів зі спеціалізованим досвідом керування ІТ-проектами; ускладнення через несумісну технологію або несправне встаткування; повна відсутність розуміння особливостей реалізації ІТ-проектів людьми, відповідальними за їхнє планування. Як наслідок, проекти провалюються за трьома групами наступних обставин (табл 2):

Таблиця 2 Причини провалу ІТ проектів

Назва причини	Опис
<i>помилки планування</i>	
невиконання строків	день здачі проекту підходить і доводиться попереджати замовника, що проект не буде готовий
проект закритий	після множинних затримок та перенесення строків замовник відмовляється від проекту
зміна цілей	проект завершений, але проблема, для розв'язку якої він був розроблено декілька місяців тому, замінена більш насущною проблемою
досягнення невірних цілей	програмний продукт має безліч потенційно цікавих можливостей, які було розумно реалізувати, однак жодна з них, і навіть їх сукупність не є тим, що принесе багато користі в результаті вирішення поставленого завдання
<i>зростання масштабу завдань</i>	
неможливість розвитку проекту	проект успішно експлуатується, але через кілька років виникає необхідність змін у системі та кількість відповідних помилок із-за змін зростають настільки, що програму необхідно міняти
змінність персоналу	після двох років розробки всі кваліфіковані працівники знаходять більш значні проекти та заробітну платню
<i>неякісне програмування</i>	
рівень помилок	проект завершений, але рівень помилок настільки високий, що програма так і не доходить до практичного використання
нерозуміння цілей	проект завершений, однак він не вирішує поставлених питань

Такі методи, як створення єдиних показників для виміру продуктивності проекту й стандартизація використовуваної для їхнього опису мови, є основними ідеями управління проектами, здатними значно поліпшити результати ІТ-проектів. Ретельне спостереження за строками й вартістю проекту й керування масштабом проекту вкрай необхідні для втримання ІТ-проектів від вищеперелічених ризиків. Маючи в якості виконавців в основному лише молодих спеціалістів, можна із упевненістю сказати, що зростуть практично всі ризики, пов'язані з розробкою програмного забезпечення. Враховуючи ж високу плинність кадрів, можна зробити висновок, що особливо виростає ризик «звільнення персоналу», з усіма, що випливають із цього наслідками, а саме: затримка здачі проекту; додатковий період адаптації нових кадрів, що приводить до додаткових затримок; складний, неочевидний дизайн програмних кодів; повний «колапс» проекту, коли фахівець пішов, не залишивши вихідних кодів або залишив вихідні програми, у яких ніхто нічого не може зрозуміти.

Практика показує, що «колапс» проекту у випадку звільнення одного конкретного фахівця, незважаючи на своєрідність ситуації, більш ніж реальна й така ситуація зустрічається досить часто.

Природно, однієї з основних цілей будь-якого управління проектом є, в першу чергу, зниження ризиків. У випадку ж, коли ризики високі, керуванню ІТ

проектом і правильної організації робіт над проектом слід дати особливий пріоритет. Одним зі способів такої організації робіт над проектом, який має на меті зібрати більшу частину ризиків, є екстремальне програмування (Extreme Programming).

Деяким людям принципи екстремального програмування нагадують звичайний здоровий глузд. Воно ґрунтується на принципах здорового глузду й застосовує їх до надзвичайного рівня [14].

Extreme Programming робить два набори «обіцянок»:

1. Програмістам обіцяє, що вони щодня будуть працювати над речами, що дійсно мають значення. Їм не прийдеться зустрічатися з непередбаченими ситуаціями поодиночі. Вони будуть здатні робити все можливе для того, щоб зробити їхню систему успішною. Вони будуть ухвалювати рішення, які можуть робити найкраще й не навпаки.

2. Замовникам і керівникам проектів обіцяє, що вони одержать максимально можливий ефект від кожного тижня програмування. Кожні кілька тижнів вони будуть здатні бачити конкретне просування до мети. І, що особливо важливо, вони будуть здатні змінити напрямок проекту в середині його розвитку без непомірних витрат на це [14].

Таким чином, Extreme Programming обіцяє зменшити ризик проекту, поліпшити можливості відгуку на зміну цілей, підвищити продуктивність робочого процесу протягом усього циклу життя системи й додати інтерес до створення програмного забезпечення в командах - усе це одночасно.

Розглянемо основні техніки Extreme Programming [7, 8]: гра планування - визначається масштаб наступного релізу, поєднуючи вартість завдання й технічні оцінки. Якщо реальність не відповідає плану, план модернізується; маленькі релізи - швидко розроблюється проста система, потім випускаються нові версії через дуже короткі інтервали; метафора - направляє всю розробку системи простим доступним описом того, як уся система працює; простий дизайн - система розробляється настільки просто, наскільки можливо в будь-який даний момент часу. Додаткова складність віддаляється, як тільки вона буде виявлена; тестування - програмісти безупинно пишуть тести модулів, які повинні проходити бездоганно для того, щоб розробка тривала далі. Замовники пишуть тести, що демонструють, що функціональні особливості закінчені; рефакторинг - програмісти реструктурують систему, не змінюючи її поведінку, з метою видалення дублювання, поліпшення зв'язків, спрощення або додавання гнучкості; парне програмування - увесь програмний код пишеться двома програмістами на одній машині; колективна власність - кожної може змінити будь-який код у системі в будь-який час; безперервна інтеграція - як тільки якесь завдання реалізоване, воно відразу ж інтегрується в систему, інтеграції відбуваються багато раз у день; доступність замовника - у команду включається справжній користувач, здатний відповісти на запитання протягом усього робочого тижня; стандарти кодування - з метою поліпшення спілкування між розроблювачами на рівні програмного коду програмісти пишуть увесь код відповідно до прийнятих правил.

Висновки. Узагальнюючи ці техніки разом щодо розробки в стилі Extreme Programming можна сказати, що слід досліджувати особливості командної роботи, а саме коли: пари програмістів працюють разом; розробка супроводжується безперервним тестуванням; пари програмістів не тільки придумують тести й

змушують їх працювати; інтеграція негайно іде за реалізацією, включаючи тестування самої інтеграції.

Парна робота за одним комп'ютером, безперервне тестування, постійна доробка дизайну й коду (рефакторинг), інтеграція відразу після реалізації нової функціональності - усі ці техніки проектного менеджменту та Extreme Programming разом покликані, в першу чергу, для того, щоб і дизайн, і сам код програми було легко модифікувати в будь-який час життя проекту, як на етапі первинного проектування системи, так і через роки після запуску програми в промислову експлуатацію.

У свою чергу, колективна власність на програмний код дозволяє програмістам швидше підвищувати кваліфікацію. Крім того, загальний програмний код повністю виключає можливість виникнення ситуації, коли один із програмістів іде й несе із собою частину коду, роблячи роботу іншого неможливою, а тим самим довго розроблювальний і можливо майже закінчений ІТ-проект може загинути. Насправді, колективна власність на код і парне програмування разом повністю виключають можливість такої ситуації, бо команди обов'язково тією чи іншою мірою будуть розбиратися у всіх кодах програми, що дозволить їм швидко зрозуміти навіть ті ділянки коду, які розроблялися переважно кимось іншим у команді.

Таким чином, коли розробкою програмного забезпечення займаються переважно малокваліфіковані фахівці, застосування сучасних методик управління програмним проектом стає особливо актуальним.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Гейтс Б. Дорога в будущее / Б. Гейтс. – М. : Изд. отд. «Русск. ред.» ; ТОО «Channel Trading Ltd», iBooks, 1996. – 312 с.
2. Дороніна М. С., Наумік К. Г. Організаційна культура //Харків: ХНЕУ. – 2006. – 86 с.
3. Друкер Питер Ф. Посткапиталистическое общество / П. Друкер. – М. : Интерсоциоинформ, 1995. – 239 с.
4. Друкер Питер Ф. Эффективный руководитель М.: Манн, Иванов и Фербер, 2013. – 349 с.
5. Иноземцев В. Л. Современное постиндустриальное общество: природа, противоречия, перспективы / В. Л. Иноземцев. – М. : Логос, 2000. – 304 с.
6. Итоги рейтинга «Информационные технологии для промышленности - 2013» [Электронный ресурс]. - Режим доступа: <http://www.up-pro.ru/specprojects/it-rating/itogi-2013.html>
7. Каштанов, Д. Постоянная техническая задолженность // [Электронный ресурс]. - Режим доступа: http://mr2dark.blogspot.ru/2009/03/blog-post_21.html
8. Мариничева, М. 10 общепринятых заблуждений об управлении знаниями (Knowledge Management) [Электронный ресурс]. - Режим доступа: http://www.iteam.ru/publications/human/section_55/article_3080
9. Наумік Е. Г., Нечепуренко А. И. Понятие «социализация» как основа формирования социального капитала предприятия //Научно-технический сборник. Коммунальное хозяйство городов. – 2006. – №. 71. – С. 325-329.

10. Основные понятия представления знаний [Электронный ресурс]. - Режим доступа: <http://itteach.ru/predstavlenie-znaniy/osnovnie-ponyatiya-predstavleniya-znaniy>
11. Рыжков, Е. Программный код и его метрики [Электронный ресурс]. - Режим доступа: <http://habrahabr.ru/company/intel/blog/106082/>
12. Чугреев В.Л. Организационные аспекты управления знаниями // Научный журнал Вопросы территориального развития, № 2, 2013. С. 1-8 // [Электронный ресурс]. - Режим доступа: <http://cyberleninka.ru/article/n/organizatsionnye-aspekty-upravleniya-znaniyami-v-it-kompanii>
13. Bell D. The coming of post-industrial society: a venture in social forecasting / D. Bell. – New York : Basic books, 1999. – 507 p.
14. Cultural Transformation <http://mka-charlotte.com/cultural-transformation/award-of-the-year-culture/>

УДК 703.5:30.022

ДЕРЖАВНА ПІДТРИМКА СІЛЬСЬКОГОСПОДАРСЬКИХ ТОВАРОВИРОБНИКІВ

Оруджева Т.М. – магістр,

Рябенко Г.М. – к.е.н., доцент,

ДВНЗ «Миколаївський національний аграрний університет»

У статті визначено оцінку сучасного стану державної підтримки сільськогосподарських товаровиробників. Висвітлено основні причини її низької ефективності, а також проаналізовано механізм державного регулювання сільського господарства країн-членів Європейського союзу.

Ключові слова: сільськогосподарське товарне виробництво, тваринництво, державна підтримка галузі, фінансове забезпечення, бюджетні дотації, аграрна політика ЄС.

Оруджева Т.М., Рябенко Г.Н. Государственная поддержка сельскохозяйственных товаропроизводителей

В статье определено оценку современного состояния государственной поддержки сельскохозяйственных товаропроизводителей. Освещены основные причины ее низкой эффективности, а также проанализирован механизм государственного регулирования сельского хозяйства стран-членов Европейского союза.

Ключевые слова: сельскохозяйственное товарное производство, животноводство, государственная поддержка отрасли, финансовое обеспечение, бюджетные дотации, аграрная политика ЕС.

Orudzheva T.M., Ryabenko G.M. State support of agricultural producers

The article defines the current state of state support of agricultural producers. The main causes of its low efficiency, and analyzes the mechanism of state regulation of agriculture of member countries of the European Union.

Keywords: agricultural commodity production, livestock production, government support of the industry, financial support, budgetary subsidies, agricultural policy of the EU.

Постановка проблеми. Загострення кризових явищ в АПК вимагають розроблення дієвої аграрної політики на основі механізму ефективної державної підтримки сільськогосподарських товаровиробників. Державна підтримка вітчиз-