

УДК 004.5

## **ІНФОРМАЦІЙНО-КОНЦЕПТУАЛЬНА МОДЕЛЬ ВЕБ-РЕСУРСУ «КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ» НУ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»**

**Агієнко М.Ю.**, здобувач вищої освіти гр. ПІ-171

**Неруш К.Ю.**, здобувач вищої освіти ПІ-171

Науковий керівник: **Трунова О.В.**, к.пед.н., доцент

*Національний університет «Чернігівська політехніка»*

Основними завданнями нової парадигми освіти в сучасних інформаційних умовах є розробка і впровадження цілісного та інтеграційного підходів і способів освоєння світу, а також забезпечення доступу до отримання освіти та безперервного підвищення культурного, освітнього та професійного рівня без обмежень. Освітня установа є відкритою системою та потребує створення особливого інформаційного середовища для забезпечення ефективної взаємодії всіх учасників освітнього процесу. Подібним майданчиком інтернет-спілкування здобувачів вищої освіти (ЗВО), батьків, науково-педагогічних працівників (НПП) та стейкхолдерів може виступати веб-сайт [1].

Сайт кафедри – інформаційна, наукова, освітня та комунікаційна основа модернізації педагогічної діяльності кафедри ВНЗ, що сприяє створенню нової моделі освіти в умовах інформатизації системи освіти суспільства.

Інформаційно-концептуальна модель формує основні параметри майбутнього веб-ресурсу, що включає критерії підбору та розташування матеріалу, а також сприяє максимальному донесенню інформації до користувача. Основою для створення веб-ресурсу слугувала інформація, яку надала кафедра «Інформаційних технологій та програмної інженерії» НУ «Чернігівська політехніка», що включала текстовий та ілюстративний матеріал.

Основні завдання сайту кафедри це:

–забезпечення своєчасного доступу ЗВО та НПП до постійно оновлюваної інформації про педагогічну діяльність кафедри та університету;

–забезпечення вільного доступу ЗВО, батькам, НПП та стейкхолдерам до нормативно-правових документів, що регламентують навчальну діяльність (навчальні плани, освітні програми, робочі програми, що відповідають кваліфікаційним вимогам підготовки фахівця та ін.);

–ознайомлення з процесом, формами та методиками навчання, а також науковою та дослідницькою діяльністю, забезпечення ЗВО та НПП доступу до всіх наукових, методичних матеріалів та комп'ютерних засобів навчання ВНЗ;

–створення умов для впровадження інноваційних технологій навчання в педагогічну діяльність кафедри (чати, електронні семінари, технології дистанційного та комп'ютерного навчання);

–концентрація на сайті кафедри посилань на тематичні освітні сайти і портали інших ВНЗ;

–надання фотоматеріалів про діяльність кафедри, НПП, робіт ЗВО і випускників кафедри, ведення поточної інформаційної та рекламної роботи;

–забезпечення безперервного моніторингу наукової та педагогічної діяльності кафедри.

При створенні структури веб-ресурсу, яка розділяється на внутрішню та зовнішню, формується перелік сторінок та взаємозв'язок між ними з урахуванням зручності орієнтації та пошуку інформації.

Розроблена внутрішня структура веб-ресурсу включає в себе весь контент розбитий на основні теми (розділи), яка враховує продумані логічні зв'язки між ними (рис. 1).

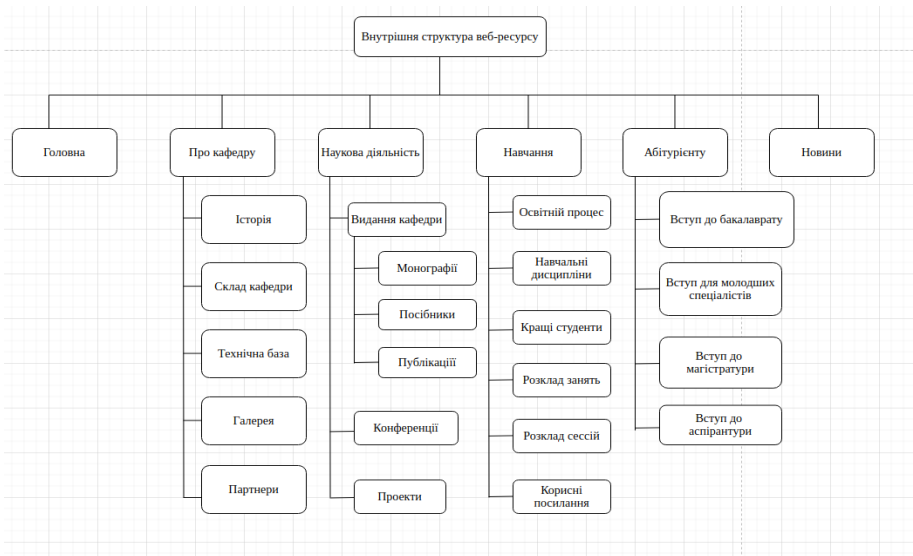


Рисунок 1 - Внутрішня структура веб-ресурсу «Кафедра ІТтаІІ»

Розроблена зовнішня структура веб-ресурсу, яка відповідає за розташування основних важливих елементів на сторінці, що включає в себе логотип, меню, пошук, інформаційні блоки, контактну інформацію тощо (рис. 2).

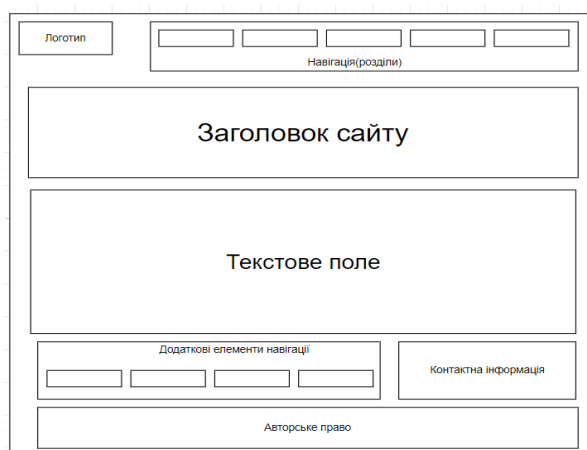


Рисунок 2 - Зовнішня структура веб-ресурсу «Кафедра ІТтаІІ»

Практичний досвід показує, що активно працюючий освітній веб-сайт дозволяє успішно вирішувати такі важливі завдання, як просування освітніх послуг, підтримка іміджу кафедри, комунікації з професійним і науковим співтовариством, супровід і підвищення комфортності освітнього процесу, розвиток взаємовідносин з учасниками ринку освітніх послуг і ринку праці, залучення спонсорів та партнерів [2]. Створення освітніх сайтів на кожній кафедрі вузу, на нашу думку, дозволить змінити ставлення до процесу навчання і сприятиме впровадженню технологій комп'ютерного та дистанційного навчання.

### Список використаних джерел

1. Березко О.Л. Пелешин А.М., Жежнич П.І. Концепція створення веб-сайта Національного університету „Львівська політехніка” / О.Л. Березко, А.М. Пелешин, П.І. Жежнич [Електронний ресурс]. – Режим доступу: <http://science.lpnu.ua/sites/default/files/journal-paper/2017/jun/3146/12-berezko-57-65.pdf> (дата звернення 19.12.2020).

2. Пелешишин А.М. Формування суспільного авторитету ВНЗ шляхом комплексного подання в системі Вікіпедія / А.М. Пелешишин, Ю.Й. Пероганич, О.В. Марковець, Н.О. Думанський // Вісник Нац. ун-ту «Львівська політехніка». – 2009. – № 653. – С. 180-187

УДК 004.415

## РЕФАКТОРИНГ ЯК ПРОЦЕС УДОСКОНАЛЕННЯ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ

**Ладоха А. В.**, здобувачка вищої освіти 3 курсу  
**Петренко Д. О.**, здобувач вищої освіти 6 курсу  
Науковий керівник: **Білоус І.В.**, доцент, к.т.н  
*Національний університет «Чернігівська політехніка»*

Рефакторинг (або його інколи називають реорганізацією коду) – це процес зміни внутрішньої структури програмного продукту, який не зачіпає його зовнішню поведінку й має на меті полегшення розуміння програмного коду й оптимізацію продуктивності. Рефакторинг повинен відбуватися постійно під час розробки програмного продукту. Навіть якщо на початку розробки код був ідеальний, в процесі розробки повинен відбутися рефакторинг, який удосконалює якість програмного продукту. Без рефакторингу не обходиться жоден дійсно складний і довготривалий проект. Адже якщо цей проект постійно використовується, то він постійно дороблюється, відповідно з'являється додатковий код, який може призвести до суцільного хаосу, якщо не буде проведено рефакторинг.

Якщо розробники хочуть мати добре структурований код, який легко читається та швидко допрацьовується, тоді слід постійно проводити рефакторинг. Код, який пройшов рефакторинг, тобто «чистий код», значно легше й дешевше підтримувати, ніж заплутаний код [1]. Але рідко вдається мати такий код постійно, адже розробники поспішають, у процесі можуть змінюватися вимоги до поставленої задачі, тестувальники можуть знаходити помилки в роботі програми, які потрібно швидко виправити, або виникають термінові доопрацювання, які доводиться робити нашвидкуруч.

Як зрозуміти, що коду потрібен рефакторинг. У першу чергу, «чистий код» повинен проходити всі тести, навіть якщо проходить 95% тестів, значить десь існує «брудний код». Код, який не потребує реорганізації, повинен бути очевидним для інших розробників. Погане іменування змінних, завеликі класи й методи – це все впливає на очевидність коду. Постійне дублювання коду – це також є ознакою наявності «брудного коду». Йї останньою ознакою «брудного коду» є кількість класів та інших рухомих частин. Чим менше класів та й самого коду, тим менше вірогідність того, що програмний продукт може «зламатися» при черговій зміні коду. Звісно, якщо код занадто заплутаний і потребує великої кількості спроб рефакторингу, простіше буде написати програмний продукт з нуля.

Яким чином проводиться рефакторинг. Під час рефакторингу проводяться маленькі послідовні покращення коду. У заплутаному програмному коді, можна чистити все, але варто почати в першу чергу з деяких проблем [2]:

1. Мертвий код. Видаленню підлягають змінні, методи або класи, які не використовуються й навряд чи знадобяться. Гірше всього, якщо мертвий код зустрінеється в умовно складній конструкції, яка не виконується через помилку або зміну вимог.

2. Дублювання. Якщо один й той самий код зустрічається не один раз у програмі, то його потрібно винести в окрему функцію.

3. Неправильні назви змінних, функцій, класів, які не передають суті їх призначення. Імена повинні повідомляти, навіщо даний елемент коду існує, що він робить і як