

$$\sum_{i=1}^n h_i x_i \rightarrow \min, \sum_{i=1}^n v_{ik} x_i \geq w_k, k = 1, 2, \dots, p, \sum_{i=1}^n x_i = 1.$$

Такі задачі як правило розв'язують методами лінійного програмування.

Проте, для використання в практичній діяльності часто важко формалізувати та застосувати той чи інший метод та модель з багатьох причин.

Тому актуальним є створення програмної системи, яка б забезпечила можливість автоматизувати процеси математичних розрахунків та надати можливість технологу працювати з системою у звичній для нього термінології.

## II. Мета роботи

Метою розробки є підвищення ефективності процесів формування та оптимізації складу композитних сумішей на основі створення програмної системи.

## III. Особливості програмної системи для оптимізації складу композитних сумішей

У роботі запропоновано створити програмну систему на основі використання веб-сервісів та бази даних про наявні матеріали та їх характеристики. Система розроблена на мові програмування php на основі Yii-framework.

Всі дані зберігаються в базі даних, розробленій на MySQL і з якою працюють сервіси.

## Висновок

Розроблено веб-орієнтовану програмну систему для оптимізації складу композитних сумішей, яка може бути використана в різних предметних областях.

## Список використаних джерел

1. Акулич И.Л. Математическое программирование в примерах и задачах. – 3 изд. – СПб.: Лань, 2011. – 319 с.
2. Дивак М.П. Оптимальна адаптивна процедура прийняття рішень на основі інтервальних моделей. / М.П. Дивак, А.В. Пукас, Г.В. Сапожник // Відбір і обробка інформації. – 2006.-№100.-с.15-21.

УДК 004.415.53

# АВТОМАТИЗАЦІЯ ПРОЕКТУВАННЯ ТЕСТІВ НА ОСНОВІ ВИМОГ

**Рачок В.С., Черешнюк О.А.**

*Тернопільський національний економічний університет, магістранти*

## I. Постановка проблеми

Розробка програмних систем, повинна супроводжуватися процесом тестування. Стандарти, що регламентують розробку програмних систем, розглядають необхідність тестування системи з використанням вимог, в тому числі представлених у формі документів. При цьому, використання документів у початковій формі може ускладнювати процес розробки через нечітку формалізацію і можливу наявність дублювання і протиріч [1, 2]. Виникає завдання аналізу і виділення окремих вимог з тексту, які згодом можуть бути використані для побудови систематизованого набору вимог [2].

Після отримання набору вимог стає можливим процедура проектування тестів. Під проектуванням тестів розуміється опис тестових сценаріїв в рамках яких може здійснюватися перевірка окремих вимог. При цьому можуть бути визначені умови, що перевіряються, а також вказані вхідні і вихідні дані.

Після етапу проектування тестів відбувається етап розробки тестових наборів, під час якого для виділених тестових сценаріїв відбувається опис процедури перевірки вимог в зазначених умовах.

Після розробки тестових наборів стає можливим проведення їх виконання, після обробки результатів якого можна стверджувати про відповідність системи вимогам в рамках наявного набору спроектованих і розроблених тестів. Також можна розглядати питання покриття набору вимог тестовими сценаріями і безпосередньо тестами.

Розглянутий процес тестування може бути проведений з різними формами представлення вимог. У даній роботі буде розглянуто використання текстового представлення. Подібне рішення, зокрема, пов'язано з наявністю великої кількості документів, приведення яких у більш формалізоване представлення вимагає значних витрат.

Розглянутий підхід до тестування дозволяє виділити життєвий цикл вимоги в випадку відсутності змін (Рис. 1).

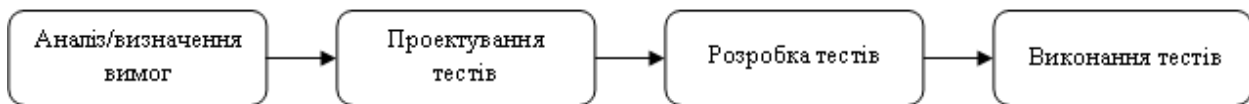


Рисунок 1 - Життєвий цикл тестів

При розгляді життєвого циклу тестів, можна виділити ряд проблем, пов'язаних як з процесом проектування тестів, так і з підтримкою вже наявного набору вимог і тестових сценаріїв.

Перша з них – отримання структурованого набору вимог з наявної документації. Як було згадано вище, проблема може бути розділена на виділення вимог і аналіз їх взаємовідношень.

Наступною проблемою є відстеження змін, в тому числі як у вихідних документах, так і в наборі вимог і тестових сценаріїв. Можна відзначити, що вимоги, в тому числі описані в формі документів, час від часу змінюються. Також можливі зміни набору вимог і тестових сценаріїв. У зв'язку з цим виникає необхідність у підтримці актуальності стану всіх елементів, що беруть участь у проектуванні тестів.

Третя проблема може бути позначена як підтримка набору схожих тестових сценаріїв. Виникають ситуації, при яких необхідно визначати і підтримувати набір артефактів (зокрема, тестових сценаріїв та вимог) опис яких є схожим, відрізняючись рядом параметрів. У ролі параметрів можуть виступати набори значень, в тому числі числових.

Остання з розглянутих проблем полягає в підтримці подібних фрагментів набору артефактів для декількох програмних продуктів або різних версій одного продукту. За підтримки набору вимог для різних версій одного продукту, деякі фрагменти набору вимог і спроектованих тестів можуть бути використані для інших версій даного продукту.

## II. Аналіз програмних засобів в області підтримки вимог і проектування тестів

Було розглянуто ряд наявних програмних засобів, використовуваних як в області аналізу і проектування вимог, так і безпосередньо для проектування тестів. Були обрані як комерційні продукти, так і вільне програмне забезпечення.

Комерційні засоби:

- IBM Requisite Pro [3];
- IBM DOORS;
- IBM Rational Requirements Composer [4];
- Caliber Requirements Management [5];
- RTIME;
- Accompa.

Вільне ПЗ:

- Open Do Qualifying Machine;
- Eclipse RMF – ProR;
- Requality.

Для програмних засобів в області підтримки вимог і проектування тестів функції повторного використання об'єктів реалізовані здебільшого для комерційних програмних продуктів. При цьому наявні рішення в більшості своїй не дозволяють описувати параметризовані об'єкти. Варто відзначити, що в той же час параметризація широко використовується в окремих засобах, призначених для генерування тестових наборів. Окремо слід відзначити інструмент Requality, який містить набір функцій, який можна порівняти як з аналогами в області вільного програмного забезпечення, так і з рядом комерційних продуктів.

Отже метою даної роботи є застосування механізмів перевантаження і параметризації для вирішення актуальних завдань в галузі проектування тестів з реалізацією зазначених механізмів в програмному засобі Requality.

## Висновки

У даній роботі були розглянуті етапи процесу проектування тестів, проаналізовано ряд пов'язаних з даною предметною областю проблем. Було проведено порівняльний аналіз програмних засобів спрямованих на підтримку вимог і проектування тестів

Були розглянуті підходи до вирішення проблем пов'язаних з автоматизацією процесу проектування тестів, включаючи компактний опис схожих артефактів і фрагментів набору вимог для модельного ряду.

### Список використаних джерел

1. Лэффенгуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению: Унифицированный подход. Пер. с англ. — М.: Вильямс, 2002. — 448 с.
2. Вигерс К. Разработка требований к программному обеспечению. Пер. с англ. — М.: Издательско-торговый дом «Русская Редакция», 2004. — 576 с.
3. Опис програмного продукту IBM Requisite Pro [Електроний ресурс]. Режим доступу: <http://www-03.ibm.com/software/products/ru/reqpro/>, вільний.
4. Опис програмного продукту IBM Rational Requirements Composer [Електроний ресурс]. Режим доступу: <http://www-03.ibm.com/software/products/ru/trc/>, вільний
5. Caliber - learn more. Collaborative requirements and definition software [Електроний ресурс]. Режим доступу: <http://www.borland.com/products/caliber/read/>, вільний.

УДК 681.3.06+674.047

## РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОЕКТУВАННЯ ЛІСОСУШИЛЬНОЇ КАМЕРИ ЗА ДОПОМОГОЮ ІНТЕРФЕЙСУ SOLIDWORKS API

Сінкевич О.В.

*Національний лісотехнічний університет України, магістрант*

### I. Постановка проблеми

В даній роботі проведено аналіз основних принципів автоматизованого проектування лісоосушальних камер. На основі проведеного аналізу реалізовано програму, що дозволяє нам не лише визначати базові геометричні характеристики камери, але й розраховувати характеристики її компонентів. В подальшому ці характеристики використані для проектування цих компонентів.

### II. Мета роботи

Метою дослідження є розробка програмного забезпечення (ПЗ) яке повинно здійснювати автоматизоване проектування компонентів, з яких будується збірка лісоосушальної камери. Готова збірка камери необхідна нам для проведення досліджень у програмі SolidWorks Flow Simulation.

### III. Програмна реалізація

Програмне забезпечення було реалізовано за допомогою мови програмування C # в середовищі Microsoft Visual Studio 2010. Одним з головних завдань програми являється отримання і обробка вхідних даних, які вводить користувач. В подальшому результати обробки даних використовуються для проведення розрахунку, результати якого дозволяють програмі вибирати параметри компонентів лісоосушальної камери. Вхідні дані вводяться у першій вкладці головного вікна програми, вигляд якої наведено на рисунку 1. Ця вкладка дозволяє користувачеві здійснювати такі операції:

- Вибирати породу деревини;
- Вказувати відстань до країв лісоосушальної камери;
- Встановлювати висоту від штабелів до фальшстелі;
- Встановлювати висоту піддонів та відстань між штабелями;
- Задавати необхідну кількість пиломатеріалів в одному штабелі;
- Вказувати загальну кількість штабелів;
- Вибирати тип розмірів та вказувати їхні значення;
- Отримувати характеристики прокладок між пиломатеріалами;
- Здійснювати розрахунок параметрів камери згідно вхідних даних.

Програма розроблена таким чином, щоб бути простою для користувача, тому інтерфейс містить лише основні параметри, які необхідно ввести для вдалого виконання поставленого завдання. В разі необхідності користувач може скористатися довідковою системою, яка вбудована безпосередньо у програму. Окрім цього усі розміри, які надає нам програма відповідають стандартам у галузі деревооброблювальних технологій України, що дозволяє користувачеві використовувати тільки реальні розміри для проведення розрахунків та подальшого проектування камери сушіння деревини.