

УДК 378.147.88

DOI: 10.15587/2519-4984.2019.172007

ОСОБЛИВОСТІ РОЗВИТКУ М'ЯКИХ НАВИЧОК СТУДЕНТІВ ІТ-СПЕЦІАЛЬНОСТЕЙ ЗАСОБАМИ НАВЧАЛЬНИХ SCRUM ПРОЕКТІВ

О. О. Гура

У статті досліджується проблема пошуку ефективних та актуальних механізмів розвитку м'яких навичок майбутніх спеціалістів сфери інформаційних технологій у процесі їх професійної підготовки. В першій частині роботи розкривається зміст поняття «м'які навички», наводиться їх класифікація та перелік найбільш актуальних представників у професійній діяльності ІТ-спеціаліста, серед яких сучасні практики та роботодавці виділяють командну роботу, креативність, управління конфліктами, критичне мислення, презентацію, лідерство, тайм-менеджмент, прийняття рішень, комунікативні навички, гнучкість та ін. У якості підґрунтя для пошуку необхідних механізмів розвитку описаних вище навичок розглядається проектна робота, одна з найбільш поширених та популярних практик вітчизняної педагогіки, під час якої у молодих спеціалістів формується професійна самостійність та здатність до розв'язання реальних професійних задач. У процесі пошуку конкретного виду проектної роботи, акцент в якій був би зроблений на розвиток м'яких навичок, був досліджений досвід практиків і розглянуті популярні промислові методології розробки програмного забезпечення, що використовуються сучасними ІТ компаніями, серед яких виділяються дві групи – монолітні та гнучкі. Також була оцінена специфіка використання та розвитку м'яких умінь спеціалістів в процесі роботи за кожною з методологій, серед яких особливо виділялася група «гнучких» та її класичний представник – Scrum. Дослідження основних проектних ролей у даній формі організації командної роботи, таких як «власник продукту», «скрам-команда», «скрам-майстер», та процесів, таких як «формування резерву», «планування спринту», «ретроспектива» та «щоденний скрам», показало, що модель розробки Scrum найбільш повно реалізує м'які навички спеціаліста та спонукає до їх активного розвитку

Ключові слова: Scrum, м'які навички, розробка програмного забезпечення, студент ІТ-спеціальності, методологія розробки

Copyright © 2019, О. О. Гура

.This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0>).

1. Вступ

З розвитком сучасного світу задача з підготовки молодих спеціалістів сфери інформаційних технологій стає все більш комплексною і неординарною. За рахунок появи незчисленої кількості креативних стартапів та нових молодих ІТ-компаній, а також формування активної професійної спільноти програмістів, індустрія все більше відходить від монолітної корпоративної культури з її класичною ієрархією управління та розподілу обов'язків. Безперешкодна комунікація з будь-яким членом команди, пряме спілкування з замовником, активна участь у плануванні розробки та безпосередній вплив на його глобальний концепт та бачення – всі ці речі все частіше стають доступними вже з перших сходинок кар'єри, даючи простір молодим спеціалістам швидко реалізовувати свої амбіції та проявляти свою особистість.

Однак, побудований за подібними принципами ринок праці вимагає від особистості всебічного розвитку, який забезпечував би конкурентоспроможність фахівця вже на початку його професійної діяльності і дозволяв би оперувати не тільки технічними знаннями зі спеціальності, але і здатністю оперативно реагувати на вимоги ринку, безперервно займатися самоосвітою, вміти спілкуватися і брати відповідальність за результат власної діяльності.

На думку керуючого директора «Luxoft Ukraine», застаріла матеріально-технічна база університетів, відсутність практики і неактуальні дані призводять до того, що існуючі програми навчання не дають молодим фахівцям розуміння вимог бізнесу. Вона зазначила, що якісне викладання теорії фундаментальних наук в українських університетах частіше за все не підкріплено розвитком особистісних якостей, так званих soft skills, адже для того, щоб бути хорошим фахівцем у своїй галузі, на сьогоднішній день потрібно володіти ще одним типом навичок – вмінням вести переговори, працювати в команді, керувати людьми і працювати з запереченнями [1]. Згідно з опитуваннями роботодавців, вітчизняна вища освіта на сьогоднішній день не орієнтується на концепцію soft skills як основу вимог ринку праці до випускників університетів. Більшість викладачів і велика частина студентів не мають уявлення про досліджуване явище, вважаючи соціально-гуманітарні дисципліни, такі як іноземні мови і менеджмент, єдиним джерелом формування нетехнічних навичок майбутніх спеціалістів.

На сьогоднішній день, українська вища ІТ-освіта тільки знаходиться на порозі реформ, пов'язаних з орієнтацією навчання на розвиток навичок, затребуваних сучасним ринком праці, у тому числі і

м'яких навичок. Однією з найбільш поширених та популярних практик вітчизняної педагогіки, яка, на думку багатьох сучасних науковців, формує професійну самостійність та здатність до творчого розв'язання не лише навчальних завдань, але й тих, що виникатимуть у подальшій професійній діяльності в умовах реального виробництва [2], є проектна робота. Вона являє собою спосіб організації діяльності учнів, при якому навчальний матеріал і способи дій з ним не надаються викладачем, а студент під керівництвом викладача самостійно знаходить необхідні відомості, працює з різною інформацією, освоює заплановані способи дій в процесі вирішення власної чи командної проблеми.

Однак, методичні рекомендації до проведення групової роботи у більшості своїй описують загальні принципи, які не враховують особливості сфери інформаційних технологій та специфіку подальшої командної роботи її працівників. Відкритою залишається проблема пошуку актуальної форми проектної роботи, оптимізованої для аудиторії майбутніх працівників сфери ІТ та спрямованої саме на розвиток їх soft-skills.

2. Літературний огляд

Аналіз праць сучасних практиків педагогіки показує, що проблема дослідження, систематизації та розвитку нетехнічних вмінь студентів стає все більш популярною та обговорюваною. Переважна більшість вітчизняних дослідників проблем підготовки студентів ІТ-спеціальностей згодна вважати розвиток м'яких навичок необхідною складовою підготовки ІТ-фахівців [3–7]. Однак, учені не зійшлися в думці щодо найбільш ефективного механізму формування цих навичок.

Ряд авторів бачить формування soft skills в умовах активної участі студента в різноманітних гуртках, спілках, громадських об'єднаннях, наукових товариствах, студентському самоврядуванні тощо. Така діяльність допомагає вдосконалювати комунікативні навички та брати на себе відповідальність за рішення [4]. Інші сходяться на думці, що основою формування нетехнічних навичок студентів здатна стати виробнича практика та стажування [5]. Частина наукової спільноти взагалі схильна вважати, що розвиток м'яких навичок не потребує окремої форми роботи і вимагає перегляду вже існуючих практик, збагачуючи їх активними діалогами зі студентами, розглядом реальних прикладів, презентаціями та дискусіями [6].

Підсумовуючи вищесказане, сучасне наукове педагогічне співтовариство визнає та активно вивчає проблему необхідності розвитку нетехнічних вмінь майбутніх фахівців, проте проблема пошуку оптимальної методології розвитку soft skills у контексті підготовки студентів ІТ-спеціальностей, на нашу думку, ще залишається невирішеною.

3. Мета та задачі дослідження

Мета дослідження – пошук ефективної та актуальної форми групової роботи, направленої на розвиток м'яких навичок майбутніх спеціалістів у сфері інформаційних технологій.

Для досягнення мети були поставлені такі задачі:

1. Проаналізувати поняття «м'які навички». Сформулювати перелік найбільш актуальних м'яких навичок у професійній діяльності спеціаліста сфери інформаційних технологій

2. Дослідити популярні методології проектної діяльності на сучасному ІТ-ринку на предмет реалізації м'яких навичок учасника. Детально розглянути процеси та командні ролі найбільш релевантної методології.

3. Проаналізувати використання м'яких умінь в процесі виконання кожного з етапів роботи та обґрунтувати ефективність обраної методології у розвитку м'яких навичок майбутніх спеціалістів у сфері інформаційних технологій.

4. Класифікація м'яких навичок спеціалістів ІТ

Навички фахівця майже кожної професії можна поділити на два різновиди – hard skills (тверді навички) та soft skills (м'які навички). Під hard skills розуміються професійні, технічні вміння, що пов'язані з діяльністю в сфері формалізованих технологій. Вони є однозначними, чітко сформульованими та повністю залежними від професійної сфери їх вживання. М'які навички – комплекс неспеціалізованих, «надпрофесійних» навичок, які відповідають за успішну участь у робочому процесі, високу продуктивність і, на відміну від спеціалізованих навичок, не пов'язані з конкретною сферою вживання. До даної групи належать організаційні, презентаційні, комунікативні, управлінські та інші вміння. На відміну від вузькоспеціалізованих hard skills, м'які навички є універсальними і допомагають професійно розвиватися і будувати кар'єру в будь-якій галузі.

Оскільки для різних видів діяльності пріоритетними є різні види soft skills, їх сталого переліку або вичерпної класифікації, що покриває усі галузі, не існує. Однак, зазвичай у літературі превалюють три категорії м'яких навичок [3]:

1. Соціально-комунікативні: комунікативні навички, міжособистісні навички, групова робота, лідерство, соціальний інтелект, відповідальність, етика спілкування.

2. Когнітивні: критичне мислення, навички вирішення проблем, новаторське (інноваційне) мислення, управління інтелектуальним навантаженням, навички самоосвіти, інформаційні навички, тайм-менеджмент.

3. Атрибути особистості і складові емоційного інтелекту: емоційний інтелект, чесність, оптимізм, гнучкість, креативність, мотивація, емпатія.

Очевидно, що жодні з умінь не є взаємовиключними, а планомірний розвиток кожної категорії soft skills позитивно сприяє професійним успіхам молодих спеціалістів. Однак, практики розробки програмного забезпечення та сучасні українські ІТ-роботодавці акцентують увагу на наступних якостях, які є пріоритетними в умовах сучасного бізнесу [7–9]:

1. Командна робота – здатність працювати в команді і виконувати свої завдання, поєднуючи інди-

відуальні навички з іншими членами команди для досягнення загальної мети.

2. Пошук і аналіз інформації – вміння спеціалістів знаходити, опрацьовувати та синтезувати інформацію, що надходить з різних джерел.

3. Креативність. IT-фахівці постійно шукають нові рішення для вдоволення потенційних технологічних потреб і проблем.

4. Прийняття рішень – вміння визначити найкращий шлях, що забезпечує максимальну ефективність і успішність отримання результату з мінімальними витратами особистих чи командних ресурсів.

5. Управління конфліктами – вміння домовлятися, мінімізувати конфліктні ситуації і не допустити напруженої атмосфери у команді.

6. Критичне мислення – здатність людини ставити під сумнів інформацію, що надходить зовні, а також припускати можливість власної помилки.

7. Презентація – вміння чітко сформулювати та в найбільш вигідній формі презентувати свої ідеї іншим.

8. Лідерство – здатність приймати бізнес-рішення і керувати підлеглими.

9. Комунікативні навички – здатність передавати інформацію людям чітко, просто, ефективно і зрозуміло. Налагоджена комунікація стає основою багатьох сучасних методологій розробки.

10. Гнучкість – спроможність прийняти негативні та неочікувані зміни і негайно шукати нові творчі рішення.

11. Тайм-менеджмент – набір методик і технологій, які дають можливість людині здійснювати планування особистого часу з метою підвищення ефективності та раціональності його використання.

Однак, набір необхідних м'яких навичок може відрізнитися не тільки для різних спеціальностей, але і в рамках інформаційних технологій. Через інтеграцію IT-систем в абсолютно різні сфери людського життя, самі по собі програмні продукти сильно відрізняються за своєю складністю і структурою, що, в свою чергу, вимагає абсолютно різних підходів до їх розробки. У сучасному світі розробки програмного забезпечення існує багато методологій та моделей, кожна з яких формує унікальний підхід до роботи та висуває різні вимоги до м'яких навичок її учасників. Справедливим є і зворотне ствердження: значущим фактором практичного використання та розвитку м'яких навичок спеціаліста є вибрана модель розробки.

5. Результати дослідження

5.1. Огляд моделей розробки ПЗ

Модель розробки ПЗ – структура, що систематизує різні види проектної діяльності, їх взаємодію і послідовність в процесі розробки ПЗ. Вибір тієї чи

іншої моделі залежить від масштабу і складності проекту, предметної області, доступних ресурсів та багатьох інших чинників. Вибір моделі розробки ПЗ серйозно впливає на безпосередньо робочий процес, визначаючи склад команди, вибір стратегії, графік роботи, необхідні ресурси та вимоги до учасників процесу.

На сьогоднішній день існує багато різновидів моделей та методологій розробки ПЗ, але в загальному випадку можна виділити дві категорії: монолітні та гнучкі.

Однією з класичних монолітних моделей є «водоспадна» (waterfall model), яка зараз представляє швидше історичний інтерес, бо в сучасних проектах використовується вкрай рідко. Вона передбачає одноразове виконання кожної з фаз проекту, які, в свою чергу, суворо слідує одна за одною (рис. 1). Дуже спрощено можна сказати, що в рамках цієї моделі в будь-який момент часу команда рухається лише з попередньої в наступну фазу, ігноруючи загальне бачення проекту та ні на крок не відходячи від попередньо створеного плану.

Водоспадна модель часто інтуїтивно застосовується при виконанні відносно простих завдань, де комунікація між членами команди зведена до мінімуму, а обсяг роботи та залежності заздалегідь відомі та залишаються незмінними.

Ітераційно-інкрементальна модель є фундаментальною основою сучасного підходу до розробки ПЗ. Ключовою особливістю даної моделі є поділення проекту на відносно невеликі проміжки (ітерації), кожен з яких в загальному випадку може включати в себе всі класичні стадії, властиві водоспадній моделі (рис. 2).

Як випливає з назви моделі, їй властива певна подвійність: з точки зору життєвого циклу модель є ітераційною, так як має на увазі багаторазове повторення одних і тих же стадій; з точки зору розвитку продукту (збільшення його корисних функцій) модель є інкрементальною. Довжина ітерацій може змінюватися в залежності від багатьох факторів, однак сам принцип багаторазового повторення дозволяє гарантувати, що і тестування, і демонстрація продукту кінцевому замовнику (з отриманням зворотного зв'язку) буде активно застосовуватися з самого початку і протягом усього часу розробки проекту.

Завдяки описаним вище факторам, монолітні моделі розробки ПО висувають досить низькі вимоги до нетехнічних навичок та персональних якостей членів команди. Вони вимагають від працівників лише послідовного виконання заздалегідь спланованих завдань та не приймають участі в високорівневому плануванні та адаптації до змін.

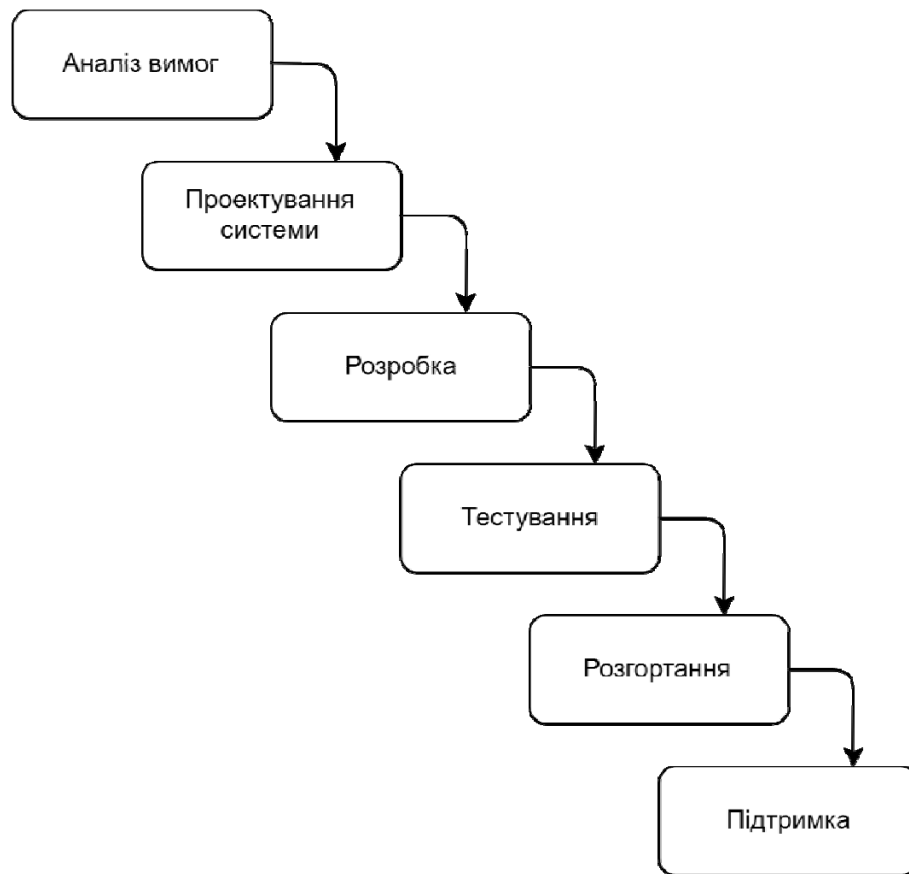


Рис. 1. Водоспадна модель розробки програмного забезпечення

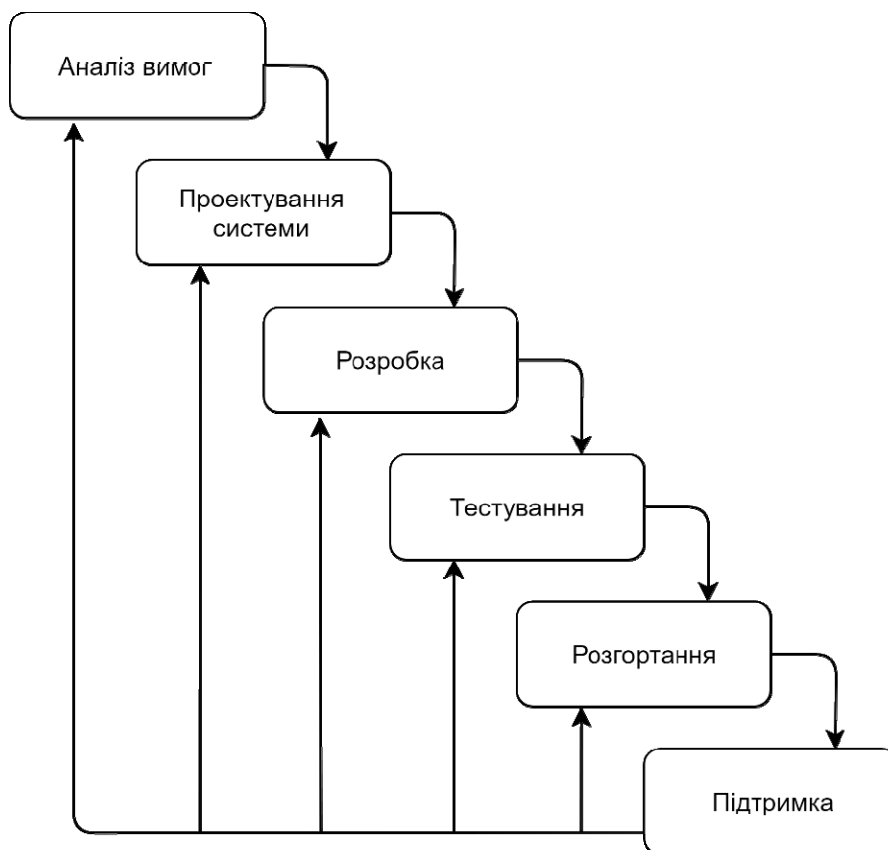


Рис. 2. Ітераційно-інкрементальна модель розробки програмного забезпечення

На відміну від класичних монолітних моделей, гнучкі методології основними пріоритетами в управлінні проектом визначають людей та їх взаємодію для створення продукту на основі співробітництва з замовником. Їх «гнучкість» забезпечується постійною відкритістю та готовністю до змін, на протидію бюрократичному дотриманню планів та постійних узгоджень монолітних моделей. Подібні технології націлені на подолання очікуваної неповноти вимог та їх постійних змін, пропонуючи гнучкий ітеративно-інкрементальний підхід в результаті постійної взаємодії команд, які самоорганізуються і складаються з фахівців різного профілю

Усі гнучкі моделі розробки ПЗ базуються на «Agile-маніфесті» – неформальному переліку основних принципів[10]:

- люди і взаємодія важливіші за процеси та інструменти;
- працюючий продукт важливіший за вичерпну документацію;
- співпраця з замовником важливіша за узгодження умов контракту;
- готовність до змін важливіша за проходження попереднім планом.

Покладені в основу гнучкої моделі підходи є логічним розвитком і продовженням всього того, що було за десятиліття створено і випробувано в монолітних моделях. Причому тут вперше був досягнутий відчутний результат в зниженні бюрократичної складової і максимальної адаптації процесу розробки ПЗ до миттєвих змін ринку і вимог замовника.

Однією з найбільш поширених та популярних гнучких методологій розробки у сучасній індустрії є Scrum (з англ. сутичка). Методологія дає змогу команді обирати завдання для виконання, враховуючи бізнес пріоритети та технічні можливості, а також вирішувати, як їх ефективно реалізувати. Це дає змогу створити умови, в яких команда працює із задоволенням і максимально продуктивно. Scrum фокусується на постійному визначенні пріоритетних завдань, ґрунтуючись на цілях бізнесу, що збільшує корисність і прибутковість проекту на його ранніх стадіях.

Оскільки за ініціації проекту його прибутковість визначити майже неможливо, Scrum пропонує концентрувати увагу на якості розроблення та до кінця кожної ітерації мати проміжний продукт, який можна використовувати, нехай і з мінімальними можливостями.

У класичному Scrum існує 3 базові проектні ролі [11]:

- «Product owner» (з англ., власник проекту) – сполучна ланка між командою розробки та замовником.

- «Scrum master» (з англ., скрам-майстер) – «службовий лідер». Його завдання - допомогти команді максимізувати її ефективність та контролювати виконання принципів scrum.

- «Development team» (з англ., команда розробки) - фахівці, які виконують безпосередню роботу над виробленим продуктом.

Першим кроком реалізації будь-якого проекту є створення бачення кінцевого продукту його власником (рис. 3). Далі власник продукту зустрічається зі скрам-майстром та командою для того, щоби надати їм інформацію про те, яким має бути кінцевий продукт у форматі «user-story» (з англ., сценарій користувача) – алгоритмів використання кінцевого продукту цільовим користувачем. Також на цьому етапі розраховуються трудовитрати для того, щоб зорієнтувати власника щодо термінів та необхідних ресурсів для виконання проекту.

Після опрацювання бачення продукту представниками замовника та скрам-майстром, всі члени проекту сортують, пріорітизують та згруповують отримані user-story для розуміння того, які елементи є обов'язковими до реалізації, а які можуть бути змінені чи відкладені у майбутньому. У результаті цього процесу створюється «backlog» (з англ., резерв) – список формалізованих та структурованих вимог та побажань щодо продукту, які впорядковані за ступенем важливості.

Основою Scrum є «sprint» (з англ., спринт) - фіксований термін, в перебігу якого виконується робота над продуктом. По закінченню спринту повинна бути отримана нова робоча версія продукту.

Наступним етапом після формування резерву є його розподіл між спринтами. При цьому слід раціонально розподілити всі елементи списку, щоб поставлені задачі можна було реально реалізувати у стислий проміжок часу.

Перед початком кожного спринту проводиться «sprint planning» (з англ., планування спринту) – нарада, на якій проводиться оцінка вмісту глобального резерву і формування «sprint backlog» (з англ., резерв спринту) – окремого списку, який містить завдання, що повинні бути виконані в поточному спринті. Кожен спринт повинен мати мету, яка є мотивуючим фактором і досягається за допомогою виконання завдань з sprint backlog.

Кожен день проводиться коротка нарада «daily scrum» (з англ., щоденний скрам), на якій кожен член команди відповідає на питання «що я зробив вчора?», «що я планую зробити сьогодні?», «які перешкоди на своїй роботі я зустрів?». Завдання daily scrum – визначення статусу і прогресу роботи над спринтом, раннє виявлення перешкод, що виникли, вироблення рішень щодо зміни стратегії, необхідних для досягнення цілей.

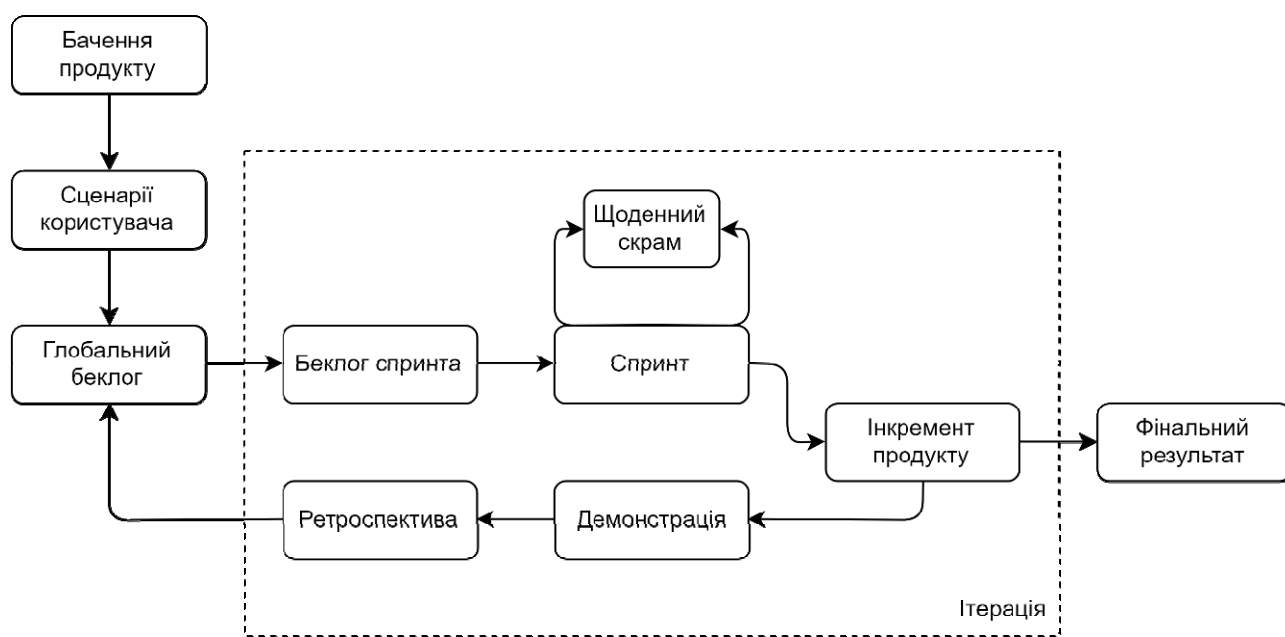


Рис. 3. Модель розробки програмного забезпечення SCRUM

Після завершення спринту команда представляє власнику результат роботи на зустрічі. Вона не є формальною, а скоріш спрямована на контроль проміжних результатів роботи над проектом.

Етап «ретроспектива», який проводиться після закінчення кожного спринту, важливий для розуміння подальших дій як команди, так і власника продукту. В ході наради відбувається аналіз проблем, з якими зіштовхнулася команда у процесі виконання спринту та пропонуються дії для запобігання подібних ситуацій у майбутньому.

Подібні ітерації (планування, розробка, демонстрація, ретроспектива) проводяться до тих пір, поки резерв завдань не буде повністю вичерпаним, а продукт буде відповідати і вимогам замовника.

5.2. Аналіз використання м'яких умінь в моделі Scrum

Задля успішної роботи Scrum – проекту потрібна команда, яка здатна до самоорганізації, тобто команда, що сама вирішує, як досягти мети. Кожен член команди повинен грати активну роль на кожному етапі, починаючи з формування ідеї, розробки про-

грамного забезпечення, тестування, і до фази впровадження та підтримки [12]. Однак, Scrum вимагає високого рівня як індивідуальної, так і командної автономії. Ця методологія розробки ПЗ найбільш повно реалізує soft-skills спеціаліста, бо ефективність і висока продуктивність команди прямо підтримуються навичками спілкування, управлінськими навичками, навичками мислення, навичками роботи в команді, лідерськими навичками – всіма найбільш актуальними м'якими навичками для працівника ІТ-сфери.

Табл. 1 яскраво ілюструє, які м'які навички найбільш активно використовуються під час кожного з етапів розробки програмного забезпечення за методологією Scrum.

Етап розробки та затвердження бачення продукту, формально, повністю виконується стороною замовника. Однак, залученість членів команди, постійні дискусії та консультації є необхідними задля складання технічно коректних та реалістичних вимог. Здатність спеціаліста до аналізу предметної галузі, креативність запропонованих рішень та гнучкість у спілкуванні з замовником є запоруками успіху у цих активностях.

Таблиця 1

Використання м'яких навичок під час scrum-активностей

Етапи проекту Scrum	Навички що використовуються	М'які навички
Бачення продукту	1, 3, 4, 10	1. Пошук і аналіз інформації 2. Командна робота
Розробка беклогу	2, 10,11,12	3. Креативність 4. Прийняття рішень
Планування спринта	2, 10,11,12	5. Управління конфліктами 6. Критичне мислення
Розробка. Щоденний скрам	1, 2, 3, 5, 9, 12	7. Презентація 8. Лідерство
Демонстрація спринта	3, 7, 9	9. Спілкування 10. Гнучкість
Ретроспектива	2, 5, 9	11. Планування 12. Тайм-менеджмент

Під час планування роботи та вибору завдань до реалізації, команда мусить діяти максимально ефективно та злагоджено. Розуміння швидкості роботи кожного з членів та усвідомлення вірогідних ризиків дозволять скрам-команді бути впевненою у подальшому виконанні своєї роботи.

Щоденна робота члена agile-проекту пов'язана з обробкою великої кількості інформації та потребою у постійній комунікації зі своїми колегами. Стислий графік розробки та нерідка поява конфліктів вимагають здатності постійно змінювати темп своєї роботи і переключатися між завданнями.

Після виконання регламентованого обсягу роботи команда повинна презентувати результати замовнику, що вимагає вміння у лаконічній та зрозумілій формі підкреслити сильні сторони продукту, а подальший аналіз вже виконаних задач на ретроспективі стане ефективним тільки завдяки здатності членів команди до відкритого та неконфліктного діалогу.

Наведені вище приклади яскраво ілюструють, що робота в скрам-команді є фактично випробувальним полігоном м'яких навичок спеціалістів ІТ - вона стимулює їх до постійного вдосконалення та еволюції.

6. Обговорення результатів дослідження

Підхід до організації навчальних проектів студентів ІТ через використання методології Scrum здатен привнести одразу декілька позитивних аспектів у сучасні методи проектної роботи студентів.

По-перше, він акумулює і зберігає усі переваги проектних методів навчання, зазначених вище. По-друге, він докорінно змінює роль викладача, який набуває нової, ще більш важливої для навчального процесу, позиції не тільки організатора, але й безпосередньо учасника діяльності студентів. Займаючи позицію власника продукту у методології Scrum, функція викладача не тільки зводиться до контролю проходження процесу та засвоєння змісту освіти, але й полягає у відігранні важливої та, насамперед, реалістичної ролі, взаємодія студентів з якою надасть їм актуальний досвід для подальшої реальної кар'єри.

Однак, найважливішим, на нашу думку, є безперервна практика та планомірний розвиток м'яких навичок студентів ІТ-спеціальностей. Систематично знаходячись у максимально приближеному до реальних умов середовищі, майбутні спеціалісти зможуть виховувати у собі навички командної роботи, прийняття рішень, презентації, планування, тайм менеджменту та багатьох інших якостей, потрібних сучасному ринку ІТ, в той час як розуміння методології зсередини та практичний досвід роботи в моделі сучасного бізнес-проекту допоможуть їм стати конкурентоспроможними кандидатами на посади молодих спеціалістів.

7. Висновки

1. Проаналізовано зміст поняття «м'які навички», розглянута їх класифікація та наведено перелік найбільш актуальних представників у професійній діяльності ІТ-спеціаліста, серед яких сучасні практики та роботодавці виділяють командну роботу, креативність, управління конфліктами, критичне мислення, презентацію, лідерство, тайм-менеджмент, прийняття рішень, комунікативні навички, гнучкість та ін.

2. Досліджені два типи популярних методологій проектної діяльності на сучасному ІТ-ринку – монолітні та гнучкі. Було з'ясовано що монолітні моделі розробки ПО висувають досить низькі вимоги до нетехнічних навичок та персональних якостей членів команди, а акцент дослідження був зроблений на найбільш поширеного представника гнучких методологій – Scrum. Були розглянуті основні проектні ролі у даній формі організації командної роботи, такі як «власник продукту», «скрам-команда», «скрам-майстер», та основні процеси, такі як «формування резерву», «планування спринту», «ретроспектива», «щоденний скрам».

3. Досліджено використання м'яких умінь в процесі виконання кожного з етапів роботи за методологією Scrum. Зіставлення переліку необхідних м'яких навичок та етапів, наведене у Таблиці 1, показало, що модель розробки Scrum найбільш повно реалізує м'які навички спеціаліста та спонукає до їх активного розвитку.

Література

1. Альхимович А. ІТ вимагають фахівців на стику галузей // Новини бізнесу. 2017. URL: <https://biz.nv.ua/experts/it-trebujut-spetsialistov-na-styke-otraslej-1601716.html> (Дата звернення: 30.06.2019)
2. Кравченко Г. Ю. Технологія організації проектної діяльності студентів вищих навчальних закладів: мат. доп. 7-ї Міжнар. наук.-метод. конф. та 105-ї Міжнар. конф. // Безпека людини в сучасних умовах. Харків: НТУ «ХП», 2015. С. 138–146.
3. Коваль К. Розвиток soft skills студента – один важливих чинників працевлаштування // Вісник Вінницького політехнічного інституту. 2015. № 2. С. 162–167.
4. Федорова О. В. Формирование hard skills, soft skills и digital skills у студентов факультета информационных технологий УВО «Университет управления «ТИСБИ» // Научноинформационный журнал Вестник «ТИСБИ». 2018. № 2. С. 335–341.
5. Taylor E. Investigating the Perception of Stakeholders on Soft Skills Development of Students: Evidence from South Africa // Interdisciplinary Journal of e-Skills and Lifelong Learning. 2016. Vol. 12. P. 1–18. doi: <http://doi.org/10.28945/3412>
6. Раицкая Л., Тихонова Е. Soft skills в представлении преподавателей и студентов российских университетов в контексте мирового опыта // Вестник РУДН. Серия: психология и педагогика. 2018. Т. 3, № 3. С. 350–363. doi: <http://doi.org/10.22363/2313-1683-2018-15-3-350-363>
7. Максудова Э. Soft skills для IT. Что важнее: знать Java или уметь договариваться? 2018. URL: <https://vc.ru/flood/39199-soft-skills-dlya-it-chto-vazhnee-znat-java-ili-umet-dogovarivatsya> (Дата звернення: 30.06.2019)

8. Шимкович В. Как грамотно «воспитывать» джуниоров. 2015. URL: <https://dou.ua/lenta/articles/raising-juniors/> (Дата звернення: 30.06.2019)
9. Львова А. Зачем инженеру soft skills. 2018. URL: http://habr.com/ru/company/epam_systems/blog/418621/ (Дата звернення: 30.06.2019)
10. Beck K., Beedle M., Bennekum A., Cockburn A., Cunningham W., Fowler M. et. al. Manifesto for Agile Software Development. 2001. URL: <http://www.agilemanifesto.org/> (Last accessed: 30.06.2019)
11. Schwaber K., Sutherland J. The Scrum Guide. 2017. URL: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf> (Дата звернення: 30.06.2019)
12. Omar M., Ahmad Khasasi L., Abdullah S., Hashim L., Romli R., Katuk N. Defining Skill Sets Requirements for Agile Scrum Team Formation // Journal of Engineering and Applied Sciences. 2018. Issue 13. P. 784–789.

Рекомендовано до публікації д-р пед. наук Безкаравайна Л. В.

Received date 28.05.2019

Accepted date 13.06.2019

Published date 31.07.2019

Гура Олександр Олександрович, аспірант, кафедра педагогіки та психології освітньої діяльності, Запорізький Національний Університет, вул. Жуковського, 66, м. Запоріжжя, Україна, 69600
Email: serenity3837@gmail.com

УДК 37.026(477.87) Духнович
DOI: 10.15587/2519-4984.2019.174117

АНАЛІЗ ОСНОВНИХ ПРОБЛЕМ ДИДАКТИКИ В ПЕДАГОГІЦІ ОЛЕКСАНДРА ДУХНОВИЧА

М. П. Тимчик

На основі аналізу «Народної педагогії» (1857) та інших праць видатного закарпатського вченого Олександра Духновича розкрито основні концептуальні положення, на яких ґрунтується його педагогіка.

Доведено, що О. Духнович обґрунтував загальні основи педагогіки (визначив її предмет, указав на мету і завдання виховання), виокремив та розв'язав основні проблеми дидактики (розкрив суть навчального процесу, охарактеризував зміст освіти, закономірності, методи і принципи навчання), висвітлив роль вчителя-вихователя (характерні риси, місію та обов'язки вчителів і т. п.).

Стверджено, що в розумінні О. Духновича, яке відповідає сучасному рівню психолого-педагогічних знань, навчання – це такий різновид людської діяльності, яке передбачає взаємодію, співпрацю викладача і учня, тому воно має двосторонній характер, тобто складається з двох процесів: 1. процесу викладання (діяльність учителя); 2. процесу навчання (діяльність учня, колективу).

Докладно проаналізовано провідні принципи навчання (природовідповідність навчання і виховання; принцип виховуючого характеру навчання; «чуттєвості наставлення» або принцип наочності, доступності; принцип диференційованого підходу до навчання; принцип ненасилля; принцип науковості навчання; принцип емоційності навчання тощо) та дидактичні правила їх реалізації, які знайшли відображення і обґрунтування в спадщині просвітника.

Розкрито окремі аспекти педагогічної майстерності вчителя, який повинен знати провідні дидактичні положення і правила.

Аргументовано, що сутнісним ядром дидактики О. Духновича є принцип особистісно-орієнтованого навчання, який передбачає врахування індивідуальних задатків, вмінь і можливостей учнів, розвитку їх творчих здібностей, формування їх світогляду.

Охарактеризовано дискусійну проблему відхилень у поведінці школярів та покарань у розумінні закарпатського педагога-просвітника. Доведено, що О. Духнович в цілому виступав за гуманізацію навчально-виховного процесу через відмову від засад авторитарної педагогіки.

Проведене дослідження засвідчує, що особистісно-орієнтований підхід у поєднанні з принципом природовідповідності є найважливішим елементом педагогіки О. Духновича, яка містить всі основні структурні компоненти теорії навчання

Ключові слова: дидактика, принципи дидактики Олександра Духновича, педагогіка Духновича, особистісно-орієнтоване навчання

Copyright © 2019, М. П. Тимчик.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0>).