

О. В. МНУШКА, ас., ХНАДУ, Харків,

В. М. САВЧЕНКО, канд. техн. наук, ХНАДУ, Харків

ФОРМУВАННЯ ТА КЕРУВАННЯ КОМАНДОЮ РОЗРОБНИКІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розглянуто теоретичні та практичні питання формування та керування командою, або малою групою розробників програмного забезпечення. Проаналізовано методичні та практичні аспекти використання курсу "Групова динаміка і комунікації" для формування підгрунтя для успішної адаптації майбутніх фахівців у складі команди з урахуванням культурних, мовних, географічних особливостей. Проведено декомпозицію процесу формування команд, визначено вхідні умови та вихідні результати на кожному етапі, сформульовано основні задачі, які потрібно розв'язати в процесі формування команд та доцільно включати у процес навчання студентів. Запропоновано та обґрунтовано критерії для оцінки факторів, що впливають на процес формування команди. Ці критерії доцільно використовувати для оцінки змісту та результатів навчання студентів у курсах "Групова динаміка та комунікації" та "Професійна практика програмної інженерії". Л.: 1. Табл.: 1. Бібліогр.: 14 назв.

Ключові слова: формування та керування командою розробників; групова динаміка; комунікації; програмне забезпечення.

Постановка проблеми. У сучасному світі комп'ютери охоплюють майже усі сфери життєдіяльності людства. Завдяки ним стало набагато простіше та швидше оброблювати інформацію, зберігати її та отримувати. Загалом, для того, щоб зробити найпростіший додаток, вистачить одного єдиного програміста. При зростанні складності і збільшенні термінів розробки суттєво зростають витрати на виробництво програмного продукту, що є небажаним не тільки для роботодавця у разі продуктової моделі, так і для клієнта, якщо мова йде про аутсорсингову модель виробництва програмного забезпечення.

Проблема формування та функціонування невеликих груп розробників із відносно коротким терміном існування є однією із фундаментальних для аутсорсингової моделі. Як правило терміни існування таких груп обмежуються місяцями, кваліфікація розробників повинна задовольняти клієнта та приносити дохід програмістам й фірмам, де вони працюють.

Як правило, більшість проектів реалізується із використанням Agile-технологій, що вимагають певної кваліфікації учасників малої групи (команди), розподіл ролей та обов'язків, певну дисципліну розробки тощо [1 – 3].

Обмежений термін на формування команди та гуртування останньої навколо досягнення мети є основною для успішності проектів та бізнесу

в цілому. Визначну роль при цьому відіграють не тільки професійні навички та вміння, а й процеси групової динаміки та комунікації у середині команди.

Аналіз літератури. Agile – методологія, технологія та комплекс засобів програмного забезпечення, що використовують для *організації процесу* розробки продуктів різного призначення, в тому числі й програмних. Ключовою особливістю Agile є гнучкість для забезпечення вимог клієнта на усіх етапах існування проекту, що вимагає певної кваліфікації від розробників та їх психологічної готовності до внесення змін у "готовий" або "майже готовий" продукт [1 – 3].

Процеси формування команд та взаємодії їх членів ґрунтуються на досвіді фахівців та знаннях, отриманих у курсі "Групова динаміка та комунікації", що є обов'язковим при підготовці бакалаврів зі спеціальності 121 "Інженерія програмного забезпечення" [4, 5].

У [4] проаналізовано міжпредметні зв'язки та показано, що окрім чисто професійної підготовки, що обумовлена вимогами роботодавця та відповідними стандартами у галузі розробки ПЗ, значну роль відіграють процеси взаємодії у малій групі, що базуються на інтеграції гуманітарних, соціальних, інженерних знань та вмінь. Також показано, що під час розробки програмних систем фахівець має справу із двома типами відносин: *суб'єкт-об'єктними*, що виникають у фахівця відносно предмету його діяльності, та *суб'єкт-суб'єктними*, що обумовлені під час взаємодії із членами команди, замовниками, користувачами продукту тощо.

У [5] розглянуто підходи до формування у майбутніх фахівців позитивного ставлення до професії, виробленню навичок спілкування та інших засобів колективної взаємодії у складі малої групи, показано що в процесі підготовки особливу увагу слід звернути на особливості процесів, що відбуваються в команді розробників ПЗ особливо в аспекті міжособистісних та фахових відносин, що дозволить сформувати підґрунтя майбутньої успішності фахівця.

Ефективна комунікація вимагає вільного володіння засобами комп'ютерної техніки та сучасними хмарними сервісами Google Docs, Microsoft Office 365, Zoho Office Suite тощо [6, 7], тому навчання майбутніх фахівців в галузі комп'ютерних наук використанню сучасних інформаційно-комунікаційних технологій та формування у них відповідних навичок надає підґрунтя для майбутньої професійної успішності.

Проблеми співіснування різних учасників в процесі розробки ПЗ викладаються у курсі "IT in society" [8], який є аналогом розглянутим вище курсам. Головною метою представленого курсу є формування

багатогранного розуміння взаємодії між технологією, користувачами та дизайнерами. Показано, що для досягнення мети необхідно враховувати не тільки соціальні, але й екологічні та когнітивні знання, що впливають на кожну взаємодію між учасниками процесу. Курс використовує можливості когнітивної психології, групової динаміки, психології мови, дискурсу та міжкультурної комунікації та ін. для формування відповідних знань, умінь та навичок у спеціалістів у галузі ІТ.

У Computer Engineering Curricula 2016 [9] підготовку фахівців для роботи у команді рекомендовано віднести до блоку підготовки до професійної практики (CE-PPP Preparation for Professional Practice). Основні питання: ефективна комунікація серед професіоналів та інших різноманітних аудиторій; значення лідерської та професійної взаємодії в межах міждисциплінарної команди; професійні та етичні обов'язки практикуючих комп'ютерних інженерів та наслідки їх роботи на суспільство; розуміння сучасних проблем, протягом усього життя, правові питання, в т. ч. інтелектуальної власності; вміння керувати проектами в галузі комп'ютерної інженерії.

У Software Engineering 2014 [10] питання роботи в команді також винесені в розділ професійної практики (4.10 Professional Practice): групова динаміка та психологія роботи в команді, самопізнання; взаємодія із зацікавленими сторонами; робота в полікультурних середовищах; комунікативні навички; читання, розуміння та узагальнення; написання технічних текстів, вимог тощо; командна та групова комунікація (усно та письмово, електронною поштою, тощо).

Таким чином, проблема формування та ефективного функціонування команд розробників ПЗ є актуальною, підготовка фахівців з відповідних питань починається в коледжах та університетах, а потім продовжується протягом професійної кар'єри.

Мета статті. Аналіз процесів формування команд (малих розподілених географічно груп) розробників програмного забезпечення та процесів групової динаміки й комунікації у таких групах.

Для досягнення поставленої мети необхідно провести аналіз процесів у розподілених групах розробників із урахуванням географічних, культурних та мовних особливостей.

Формування та керування командою розробників. Проект є основою для розробки програмного забезпечення. У рамках проекту відбуваються всі етапи розробки від формування технічного завдання до випуску ПЗ на ринок. Проект має обмеження – термін розробки, етапи розробки, бажаний результат на кожному етапі та бюджет, все це в

кінцевому рахунку визначає склад та кваліфікацію членів команди розробників. На основі аналізу типових ролей у проектних командах визначимо основні проблеми, що впливають як на процес формування команд так і на їх функціонування в цілому.

Проблема № 1. Якість підбору персоналу (рекрутинг). Будь-яка розробка вимагає наявності команди професіоналів, яка може бути доволі сталою у продуктивних компаніях, або формуватися під проект конкретного замовника, що є характерними для аутсорсингових компаній. На якість цього процесу впливають об'єктивні та суб'єктивні фактори.

Об'єктивні – вимоги замовника до відповідної кваліфікації спеціаліста та технічне завдання на проект, що безпосередньо визначає технології та мови програмування.

Суб'єктивні – вміння спеціаліста представити себе у резюме (суб'єктивне бачення себе та суб'єктивна оцінка з боку спеціаліста із підбору кадрів), співбесіда та оцінка її результатів (технічну співбесіду проводять спеціалісти різного рівня та з різним баченням вимог до кандидата) [11].

Через відсутність ТЗ або не чіткого його формування, можуть виникнути ряд проблем, які можуть дуже сильно вплинути на кінцевий результат (рис. 1, витоки цієї ілюстрації у середні 60-х років 20-го сторіччя, але актуальність її все ще зберігається), зроблено на основі [13]). Ясність і чіткі вимоги допомагають гарантувати, що функціональність системи визначається користувачем, а не програмістом. Якщо вимоги сформульовані недостатньо явно, або взагалі їх немає, програмісту зазвичай самому доводиться визначати їх під час програмування. Крім того, наявність явних вимог допомагає уникати суперечок.

Вимоги допомагають визначити функціональність системи до початку програмування.

Відповідно до даних Project Management Institute [13] 15% проектів зазнають цілковитої невдачі, що означає трату грошей, репутації, робочих місць; 57% відсотків проектів завершилися не перевищивши запланований бюджет, а лише 52% проектів завершилися у відведений термін часу; також 52% проектів відчували на собі неконтрольовані зміни.

Проблема № 2. Комунікативні канали та їх якість. У даному випадку під якістю будемо розуміти повноту та релевантність інформації для обох сторін – замовника та розробника.

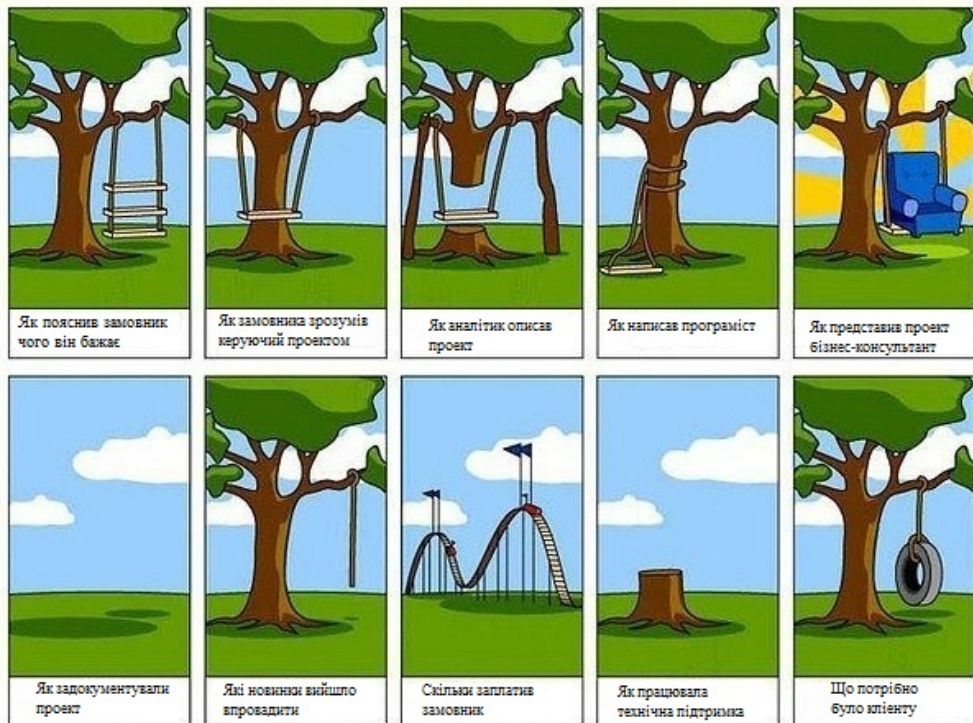


Рис. 1. Проблема комунікації розробників із замовником [12, 13]

Комунікація із замовником має бути на всіх стадіях розробки проекту [12]. Першим кроком на шляху успішного завершення проекту є визначення вимог замовника, або офіційних вимог (ТЗ – технічне завдання).

Подальший аналіз причин краху проектів показує [13], розглянуті вище проблеми групової динаміки та комунікації все ще знаходяться у лідерах, а саме: неадекватне бачення цілей проекту (29%), погана комунікація із клієнтом (29%), недосвідчене управління проектами (22%) та непродуктивне гаяння часу членами команди (13%). Зазначимо, що це характерно для всіх проектів, а не тільки для програмних.

Серед інших причин також багато питань, пов'язаних із керуванням проектом та структурою проекту, а саме: зміни в пріоритетах для фірми (39%), зміни в баченні цілей та результатів (37%), обмежені та коштовні ресурси (21%), нечітке ТЗ (35%), похибки в плануванні часу (25%) та бюджету (28%).

Таким чином, успішність проектів залежить від багатьох факторів впливу, при цьому поганий менеджмент руйнує роботу команд, призводить до неефективної роботи учасників процесу, в незалежності від їх попереднього досвіду чи кваліфікації.

Проблема № 3. Якісний та кількісний склад команди. Якісний та кількісний склад команди визначається задачами, що вирішуються, та бюджетом на проект. Як правило в складі команди приймають участь власник продукту, дизайнери, програмісти, менеджер проекту. Розглянемо типові функціональні обов'язки членів команди розробників, з точки зору впливу на формування команди.

До функціональних обов'язків програмістів (Software Developer, Software Engineer) відноситься створення програмного коду. Частина програмістів виконують роль архітекторів програмного забезпечення та визначають архітектурні шаблони, технології та парадигми, працюють із ТЗ. У складних випадках до команди також входять науковці-розробники алгоритмів, наприклад, для моделювання складних технічних або фізичних процесів, таких як переміщення роботів, маніпуляторів, комп'ютерний зір, штучний інтелект тощо. Програмісти відносяться до основного складу команди.

До обов'язків програмістів часто також відносять Unit-тестування, інколи ці задачі можуть виконувати спеціальні розробники, що мають поглиблені знання у галузі тестування (Software Engineer in Test, Software Developer in Test). До функціональних обов'язків останніх також відносять автоматизацію тестуванням, роботу із неперервною інтеграцією (Continuous Integration, CI), в залежності від складності проекту, таких спеціалістів може не бути в команді, отже це опціональна роль у команді.

До функціональних обов'язків тестувальників ПЗ (QA, Quality Assurance Engineer) відносять: оцінку програми з точки зору експерта (відповідність вимогам, помилки в роботі програм) та звичайного користувача (функціональність продукту); розробку сценаріїв тестування, тест-кейсів, їх виконання, складання звітів про виявлені помилки; мануальне тестування програмного забезпечення. Тестувальники також відносяться до основного складу команди.

До функціональних обов'язків дизайнерів (Designer) відносять розробку візуальну частину продукту – інтерфейс користувача (UI – візуальна складова, UX – поведінкова (на основі досвіду) складова). Продукт може не мати візуальної складової, тому дизайнер – опціональна роль у команді.

У залежності від проекту, є ще багато інших ролей технічних спеціалістів – системний адміністратор, DevOps, системний інтегратор, наявність таких спеціалістів є опціональною, вони можуть не бути безпосередніми членами команди, а запрошуватись для розв'язання певних задач.

До функціональних обов'язків менеджерів проекту відносять вирішення стратегічних (власник проекту/продукту або Project Owner) та тактичних (керівник проекту або Project Manager) питань. Саме керівник проекту у відповідності із домовленістю із клієнтом визначає строки завершення етапів робіт, розробляє план дій та необхідні ресурси (людські, фінансові, матеріальні тощо), керує персоналом на повсякденній основі, мінімізує ризики та організує комунікацію між усіма учасниками проекту. У невеликих командах роль менеджера проекту може виконувати керівник групи (Team Lead), але це може завадити безпосереднім обов'язкам керування командою.

Замовник (Client) – формує вимоги, фінансує розробку та користується продуктом. Замовник контролює процес розробки, вносить зміни до ТЗ, займається просуванням продукту на ринок і т.і.

Проблема № 4. Розмір команди. Члени команди мають різні професійні навички, а баланс і різноманітність навичок – ось дві найбільш важливі складові чудової команди. Особливо це важливо для малої групи фахівців, де одні здатні ефективно працювати з замовниками, а інші – мають експертизу у програмуванні певних систем або додатків, тестуванні, проектуванні архітектури програми тощо.

Проблема № 5. Формування проектної команди із окремих професіоналів. Глобалізація ринку праці чи не найпершою вплинула на ринок послуг у галузі інформаційних технологій. Аналіз розповсюдження офісів великих гравців на цьому ринку таких як Google, Amazon, Eram, SoftServe, GlobalLogic та ін. [15], показує географічну та глобальну розподіленість.

Проблема № 6. Мультикультурні та міжетнічні взаємовідносини. До звичайних проблем побудови та гуртування команди також додаються мультикультурні, мовні та етнічні особливості, та навіть святкові дні, що можуть вплинути на графік виконання робіт. Як показує аналіз типових програм підготовки фахівців в українських університетах цьому питанню приділяється недостатньо уваги, що призводить до недостатньої адаптації фахівців (і не тільки рівня Junior) до умов сучасного ринку праці.

Проведемо декомпозицію процесу формування команди розробника та визначимо основні етапи, включно із вхідними умовами та результатами на виході (табл. 1). Зазначимо, що для оцінки кожного етапу використовується декілько критеріїв, що підтверджує складність процесів, що взаємодіють в процесі формування та гуртування команди.

До звичайних проблем побудови та гуртування команди також додаються мультикультурні, мовні та етнічні особливості, та навіть святкові дні, що можуть вплинути на графік виконання робіт. Як показує

аналіз типових програм підготовки фахівців в українських університетах цьому питанню приділяється недостатньо уваги.

Розвиток передових інформаційних технологій, світова практика комплексного управління якістю, необхідність застосування інновацій потребує зміщення акценту з індивідуальної роботи окремих виконавців на діяльність багатofункціональних управлінських команд і робочих груп, сконцентрованих на комплексних проблемах і задачах. Згідно методології управління командами після розподілу ролей і відповідальності, кожний член команди має приймати участь у плануванні проекту і прийняття рішень.

Склад команди в процесі розвитку проекту не є сталим, на різних етапах до неї підключаються спеціалісти для вирішення конкретних короткотермінових задач, наприклад, налаштування служб СІ, написання драйвера пристрою тощо. Це створює додаткові виклики перед менеджментом та вимагає толерантного ставлення до новачків з боку інших членів команди. Також поширеною є практика часткової зайнятості спеціалістів.

Основні етапи створення команди та задачі, що потрібно розв'язати на цьому етапі та на етапі гуртування команди наведено у табл. 1. Зазначимо, що ця таблиця не покриває усі можливі випадки, а є загальним поглядом на проблему формування команди.

Причинами доцільності командної організації виконання проектів є їх комплексність, часто міждисциплінарний характер розв'язуваних проблем, новизна а, особливо гостра необхідність збільшення творчого потенціалу співробітників як за рахунок ефекту синергії, так і за рахунок горизонтальних зав'язків, відсутність таких стримуючих творчість факторів, як стереотипи, рутинні процедури.

Основні фактори, що впливають на формування команди:

- зовнішні фактори, такі як професійні вимоги до членів команди і відповідність рольового складу, що визначаються на етапі інноваційного макета проекту;
- внутрішні чинники, такі як соціотипи особистості претендентів в проектній команді з обмеженого числа співробітників компанії, які визначають взаємовідносини членів команди.

Таблиця 1

Декомпозиція процесу формування команди

Етап	Опис	Вхід	Вихід	Відповідальний
Аналіз необхідних ресурсів розробників.	Визначення кваліфікаційних (професійних) вимог.	Технічне завдання.	Перелік ресурсів включно із складом та кваліфікацією окремих виконавців.	Менеджер проекту.
Аналіз наявних ресурсів.	Відображення вимог на існуючий склад розробників.	Перелік ресурсів включно із складом та кваліфікацією окремих виконавців.	Перелік наявних спеціалістів включно із їх поточним завантаженням на інших проектах, перелік відсутніх ресурсів.	Менеджер проекту разом із HR (Human Resources).
Визначення вакансій та підбір персоналу.	Рекрутинг нових спеціалістів, перерозподіл наявних спеціалістів.	Перелік наявних та відсутніх ресурсів включно із складом та кваліфікацією окремих виконавців.	Укомплектована команда.	Менеджер проекту разом із HR (Human Resources), технічні спеціалісти (інтерв'ю).
Планування роботи команди.	Ознайомлення команди із технічним завданням, планом розробки, визначення відповідальних та термінів робіт.	Технічне завдання, склад команди.	План робіт узгоджений із усіма членами команди.	Менеджер проекту, керівник групи.

Продовження табл.1

Етап	Опис	Вхід	Вихід	Відповідальний
Підготовка та координація заходів щодо навчання.	Організація необхідних технічних тренінгів (onboarding).	Технічне завдання, ресурси для розробки.	Матеріали і план для технічних тренінгів.	Менеджер проекту, інструктори, система дистанційної освіти (при наявності).
Навчання співробітників.	Проведення тренінгів згідно планів та графіків.	Матеріали для технічних тренінгів.	Документи, що підтверджують проходження тренінгів та відповідну набуту кваліфікацію.	Менеджер проекту, інструктори, система дистанційної освіти (при наявності).
Гуртування команди, підтримання мікроклімату, розвиток команди.	План заходів щодо гуртування команди (team buildings).	Склад команди, аналіз професійних та людських якостей.	Команда.	Менеджер проекту.

Висновки. Проектна команда створюється на визначений термін та для вирішення конкретних задач по проекту. За час свого існування команда переживає трансформації, викликані змінами вимог з боку клієнта або закінченням певних етапів розробки. Після закінчення проекту команду можуть розпустити або частково чи повністю переключити на інший проект.

Відмінними рисами організації роботи проектною командою є: загальна мета, спільна робота, спільна відповідальність, статус членів команди, рівність її членів при плануванні задач, делегування повноважень, проектні завдання, знання і навчання в команді, інформація та комунікації, інтеграція команди в організаційну структуру підприємства, згуртованість команди.

Проаналізовано типові програми підготовки фахівців із розробки ПЗ в українських та закордонних університетах з точки зору питань організації та функціонування команд. Показано, що зміст курсу "Групова

динаміка та комунікації" в цілому відповідає викликам, що виникають при підготовці програмних інженерів. Разом з тим недостатню увагу приділено формуванню та функціонуванню мультикультурних багатоетнічних команд, що є типовим на сьогоднішній день.

Проаналізовано типові проблеми, з якими стикається менеджмент при формуванні проектної команди, проаналізовано типові ролі учасників команди, визначено основні та опціональні ролі учасників команди. Проведено декомпозицію процесу формування команд, визначено вхідні умови та вихідні результати на кожному етапі, сформульовано основні задачі, які потрібно розв'язати в процесі формування команд та доцільно включати у процес навчання студентів.

Запропоновано та обґрунтовано критерії для оцінки факторів, що впливають на процес формування команди. Ці критерії доцільно використовувати для оцінки змісту та результатів навчання студентів у курсах "Групова динаміка та комунікації" та "Професійна практика програмної інженерії".

Перспективами подальших досліджень є вивчення процесів групової динаміки та комунікації у мультикультурних багатоетнічних командах.

Список літератури:

1. Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Business Information Processing: Springer International Publishing. – 2018. – № 314. – 314 p.
2. Schmidt C. Agile Software Development Teams: the Impact of Agile Development on Team Performance / C. Schmidt // Progress in IS. – Springer. – 2016. – 184 p.
3. Гулага Я.С. Критерії оцінки якості в проектах, що використовують Agile / Я.С. Гулага, О.В. Мнушка // Комп'ютерні технології і мехатроніка. Збірник наукових праць за матеріалами міжнародної науково-практичної конференції. – Харків: ХНАДУ, 2019. – С. 82-85.
4. Кириленко Е.Г. Обоснование содержания обучения в рамках методологии преподавания профессионально-ориентированной дисциплины "Групповая динамика и коммуникация" / Е.Г. Кириленко, О.В. Лучшева // Інженерія програмного забезпечення. – 2010. – № 1. – С. 71-78.
5. Буч Ю. Навчальний курс "Групова динаміка і комунікації" в контексті підготовки сучасних фахівців у галузі інформаційних технологій / Ю. Буч // Вісник Львівського університету. Серія педагогічна. – 2017. – Вип. 32. – С. 511-519.
6. Савченко В.Н. Использование виртуализации и облачных технологий при обучении информационным технологиям / В.Н. Савченко, О.В. Мнушка, И.А. Сасимова // Нові технології навчання. – Київ: Інститут інноваційних технологій і змісту освіти МОН України. – 2016. – Вип. 88. – Частина 1. – С. 97-101.
7. Мнушка О.В. Аналіз використання хмарних технологій для формування компетенцій під час навчання в галузі інформаційних та комп'ютерних технологій / О.В. Мнушка // Вісник ХНАДУ. – Харків – 2017. – Вип. 76. – С. 123-127.
8. Computer Engineering Curricula 2016. Curriculum guidelines for undergraduate degree programs in computer engineering // IEEE Computer Society and Association for Computing Machinery. – 2016.

9. Software Engineering 2014. Curriculum guidelines for undergraduate degree programs in software engineering // IEEE Computer Society and Association for Computing Machinery. – 2015.
10. Daniels M. IT in society: a multi-disciplinary course / M. Daniels, M. Petre // FIE'99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education. Conference Proceedings. IEEE Cat. No.99CH37011, San Juan, Puerto Rico, USA, 1999. – Vol. 2. – P. 13A1/21
11. Bird C. Interviews / C. Bird // Perspectives on Data Science for Software Engineering. – 2016. – P. 125–131.
12. Meyer B. Software Architecture / B. Meyer, C.A. Furia, M. Nordio. – Software Architecture. Lecture 2: The software lifecycle. Url: http://se.ethz.ch/courser/2011a_spring/soft_arch/lectures/02_softarch-lifecycle.pdf.
13. Free download Tree Of Life png. – CleanPNG / KissPNG. URL:<https://www.cleanpng.com/png-project-management-project-manager-architectural-e-5729993/download-png.html>.
14. Success in Disruptive Times. Expanding the Value Delivery Landscape to Address the High Cost of Low Performance. Project Management Institute. – 2018. Url: <https://dou.ua/lenta/articles/top-50-jan-2019>.
15. Топ-50 ІТ-компаній України, січень 2019: зростання на 18% за рік і подолання відмітки "6000 спеціалістів". – 2019

References:

1. *Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Business Information Processing* (2018), No.314, Springer International Publishing, 314 p.
2. Schmidt, C. (2016), *Agile Software Development Teams: the Impact of Agile Development on Team Performance*. Progress in IS. Springer, 184 p.
3. Gulaga, Y.S., and Mnushka, O.V. (2019), "Quality assessment criteria in projects using Agile", *Computer technology and mechatronics. Collection of scientific works based on the materials of the international scientific-practical conference*, Kharkiv, KhNADU, pp. 82-85.
4. Kyrylenko, E.Gh., and Luchsheva, O.V. (2010), "Justification of the content of training in the framework of the teaching methodology of professionally oriented discipline "Group dynamics and communication", *Software engineering*. No.1, pp. 71-78.
5. Buch, Ju. (2017), "Training course "Group dynamics and communications" in the context of training modern specialists in the field of information technology". *Bulletin of Lviv University. The series is pedagogical* Lviv, Vyp, 32, pp. 511-519.
6. Savchenko, V.N., Mnushka, O.V., and Sasymova, Y.A. (2016), "Applying virtualization and cloud computing in the information technologies trainings", *New learning technologies*. Institute of Innovative Technologies and Educational Content of the Ministry of Education and Science of Ukraine, Kyiv, vyp. 88, chastyna 1, pp. 97-101.
7. Mnushka, O.V. (2017), "Analysis of using cloud computing for competence development in the study fields of informatics and computer technologies", *Bulletin of the KhNADU*, Vyp. 76, Kharkiv, pp. 123-127.
8. *Computer Engineering Curricula 2016*. (2016) Curriculum guidelines for undergraduate degree programs in computer engineering. IEEE Computer Society and Association for Computing Machinery. Available at: <https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf> (accessed May 2020).
9. *Software Engineering 2014*. (2015), Curriculum guidelines for undergraduate degree programs in software engineering. IEEE Computer Society and Association for Computing Machinery. Available at: <https://www.acm.org/binaries/content/assets/education/se2014.pdf> (accessed May 2020)

-
- 10.** Daniels, M., and Petre, M. (1999), "IT in society: a multi-disciplinary course", *FIE'99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education. Conference Proceedings*. IEEE Cat. No.99CH37011, San Juan, Puerto Rico, USA, 1999, vol. 2, pp. 13A1/21.
- 11.** Bird, C. (2016), "Interviews". *Perspectives on Data Science for Software Engineering*. pp. 125-131.
- 12.** Meyer, B., Furia, C.A., Nordio, M. (2011), *Software Architecture. Lecture 2: The software lifecycle*. Available at: http://se.inf.ethz.ch/courses/2011a_spring/soft_arch/lectures/02_softarch_lifecycle.pdf (accessed May 2020)ю
- 13.** Free download Tree Of Life png. – CleanPNG / KissPNG. Available at: <https://www.cleanpng.com/png-project-management-project-manager-architectural-e-5729993/download-png.html> (accessed May 2020)
- 14.** *Cost of Low Performance*. (2018)
- 15.** Top 50 IT companies of Ukraine, January 2019: growth of 18% per year and overcoming the mark of "6000 specialists".

Статтю представив д-р техн. наук, проф. ХНАДУ, зав каф. Комп'ютерних технологій і мехатроніки Ніконов О.Я.

Надійшла (received) 10.05.2020.

Mnushka Oksana, Assistant Lecturer,
Kharkiv National Automobile and Highway University,
Str. Yaroslava Mudrogo, 25, Kharkiv, Ukraine, 61002
Tel.:(+38057)707-37-47, e-mail: mnushka.ov@gmail.com
ORCID ID: 0000-0001-7756-9260

Savchenko Volodymyr, Candidate of Science (Engineering),
Kharkiv National Automobile and Highway University,
Str. Yaroslava Mudrogo, 25, Kharkiv, Ukraine, 61002
Tel.:(+38057)707-37-47, e-mail: savchenko@live.com
ORCID ID: 0000-0001-6548-0891

УДК 004.4:005.8

Формування та керування командою розробників програмного забезпечення / Мнушка О.В., Савченко В.М. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2020. – № 1 (3). – С. 99 – 112.

Розглянуто теоретичні та практичні питання формування та керування командою, або малою групою розробників програмного забезпечення. Проаналізовано методичні та практичні аспекти використання курсу "Групова динаміка і комунікації" для формування підґрунтя успішної адаптації майбутніх фахівців у складі команди із урахуванням культурних, мовних, географічних особливостей. Проведено декомпозицію процесу формування команд, визначено вхідні умови та вихідні результати на кожному етапі, сформульовано основні задачі, які потрібно розв'язати та доцільно включати у процес навчання студентів. Запропоновано та обґрунтовано критерії для оцінки факторів, що впливають на процес формування команди. Ці критерії доцільно використовувати для оцінки змісту та результатів навчання студентів у курсах "Групова динаміка та комунікації" та "Професійна практика програмної інженерії". Іл.: 1. Табл.: 1. Бібліогр.: 15 назв.

Ключові слова: формування та керування командою розробників; групова динаміка; комунікації; програмне забезпечення.

УДК 004.4:005.8

Формирование и управление командой разработчиков программного обеспечения / Мнушка О.В., Савченко В.М. // Вестник НТУ "ХПИ". Серія: Інформатика і моделювання. – Харків: НТУ "ХПІ". – 2020. – № 1 (3). – С. 99 – 112.

Рассмотрены теоретические и практические вопросы формирования и управления командой, или малой группой разработчиков программного обеспечения. Проанализированы методические и практические аспекты использования курса "Групповая динамика и коммуникации" для формирования основы успешной адаптации будущих специалистов в составе команде с учетом культурных, языковых, географических. особенностей. Проведена декомпозиция процесса формирования команд, определены входные условия и выходные результаты на каждом этапе, сформулированы основные задачи, которые необходимо решать и целесообразно включать в процесс обучения студентов. Предложены и обоснованы критерии для оценки факторов, влияющих на процесс формирования команды. Эти критерии целесообразно использовать для оценки содержания и результатов обучения студентов в курсах "Групповая динамика и коммуникации" и "Профессиональная практика программной инженерии". Ил.: 1. Табл.: 1. Библиогр.: 15 названий.

Ключевые слова: формирование и управление командой разработчиков; групповая динамика; коммуникации; программное обеспечение.

UDC 004.4:005.8

Formation and management of a team of software developers / Mnushka O.V., Savchenko V.M. // Herald of the National Technical University "KhPI". Series of "Informatics and Modeling". – Kharkov: NTU "KhPI". – 2020. – № 1 (3). – P. 99 – 112.

Theoretical and practical issues of forming and managing a team, or a small group, of software developers, are considered. Methodological and practical aspects of using the course "Group Dynamics and Communication" to form the basis for successful adaptation of future professionals in the team, taking into account cultural, linguistic, geographical. The decomposition of the team formation process is carried out, the initial conditions and output results at each stage are determined, the main tasks to be solved in the team formation process are formulated and it is expedient to include them in the student learning process. Criteria for assessing the factors influencing the team formation process are proposed and substantiated. These criteria should be used to assess the content and learning outcomes of students in the courses "Group Dynamics and Communication" and "Professional Practice of Software Engineering". Figs.: 1. Tabl.: 1. Refs.: 15 titles.

Keywords: formation and management of the development team; group dynamics; communications; Software.