

Лекція 3

Робочий продукт, дисципліна зобов'язань, проект та управління ним

Анотація: Робочий продукт. Дисципліна зобов'язань. Проект. Управління проектами.

Через творчий характер програмування, молодий вік учасників розробки ПЗ, виявляються актуальними деякі питання звичайного промислового виробництва, що стали давно загальним надбанням. Перш за все, це дисципліна зобов'язань і робочий продукт. Дані знання, коли залучені у практику, є надзвичайно корисними в командній роботі. Крім того, широко вживані зараз на практиці методології розробки ПЗ, що підтримані відповідним програмним інструментарієм, активно використовують ці поняття, уточнюючи і конкретизуючи їх.

Робочий продукт

Однією з істотних умов для керованості промисловим процесом є наявність окремо оформлених результатів роботи – як в остаточному вигляді, так і проміжних. Ці окремі результати у складі загальних результатів робіт допомагають ідентифікувати, планувати і оцінювати різні частини результату. Проміжні результати допомагають менеджерам різних рівнів відслідковувати процес втілення проекту, замовник отримує можливість ознайомитися з результатами задовго до закінчення проекту. Більш того, самі учасники проекту в своїй щоденній роботі отримують простий і ефективний спосіб обміну робочою інформацією, а саме – обмін результатами.

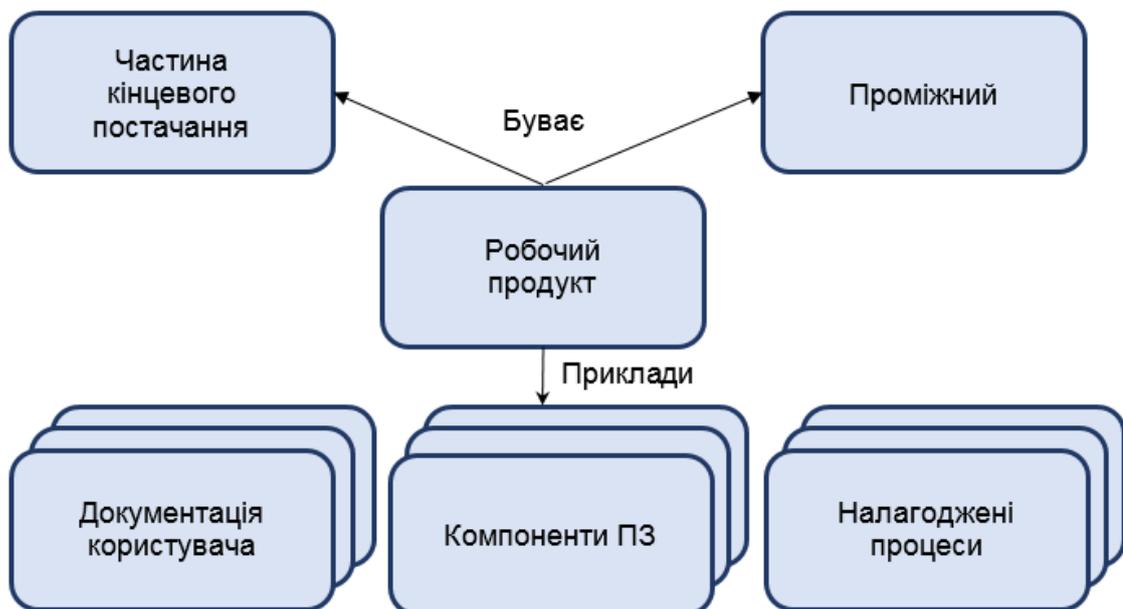


Рис. 3.1. Види типового робочого продукту

Таким результатом є робочий продукт (*work product*) – будь-який артефакт, який був створений в процесі розробки ПЗ, наприклад, файл або набір файлів, документи, складові частини продукту, *сервіси*, процеси, специфікації, рахунки і так далі (Рис. 3.1).

Ключова різниця між робочим продуктом і компонентом ПЗ полягає в тому, що перший необов'язково матеріальний і відчутний (*not to be engineered*), хоча і може бути таким. Нематеріальний робочий продукт – це, як правило, деякий налагоджений процес – промисловий процес виробництва якої-небудь продукції, учбовий процес в університеті (на факультеті, на кафедрі) і так далі.

Важливо відзначити, що робочий продукт зовсім не обов'язково є складовою частиною підсумкового постачання. Наприклад, налагоджений процес тестування системи не поставляється замовникові разом з самою системою. Уміння управляти проектами (не лише в галузі програмування) багато в чому пов'язане з мистецтвом визначати потрібні робочі продукти, наполягати на їх створенні і в їх термінах вести приймання проміжних етапів роботи, організовувати синхронізацію різних робочих груп і окремих фахівців.

Багато методології включають опис специфічних робочих продуктів, які використовуються в процесі, – *CMMI*, *MSF*, *RUP* і ін. Наприклад, в *MSF* це програмний код, *діаграми* додатків і класів (*application diagrams* *class diagrams*), план *ітерацій* (*iteration plan*), *модульний тест* (*unit test*) і ін. Для кожного з них обов'язково визначено зміст, відповідальні за розробку, *місце* в процесі та інші аспекти.

Зупинимося трохи детальніше на проміжних робочих продуктах. **Компонента ПЗ, яка створена в проекті одним розробником і надана для використання іншому розробникові, виявляється робочим продуктом.** Її треба мінімально протестувати, виправити назви класів інтерфейсів і методів, прибрати зайве, (таке, що не має *відношення* до функціональності даної компоненти) розділити на *public* *private*, і так далі. Тобто, виконати деяку додаткову роботу, яку, мабуть, розробник і не почав би робити, якби продовжував використовувати компоненту лише сам. Об'єм цих додаткових робіт суттєво зростає, якщо компонента повинна бути надана для використання в розробці, наприклад, в інший центр розробки (іноземним партнерам, що є звичною ситуацією для офшорної розробки). **Виготовлення гарних проміжних робочих продуктів є дуже важливим для успішності проекту, але вимагає додаткової роботи від їх авторів.** Працювати одному, і не надавати робочих продуктів – легше, а для багатьох і зручно. Але робота в команді вимагає накладних витрат, у тому числі і у вигляді витрат на створення проміжних робочих продуктів. Звичайно, якість цих продуктів і трудовитрати на їх виготовлення дуже варіюються залежно від ситуації, але тут важливо розуміти сам принцип.

Отже, підсумуємо, що **проміжний робочий продукт повинен обов'язково мати ясну мету і конкретних користувачів, щоб мінімізувати накладні витрати на його створення.**

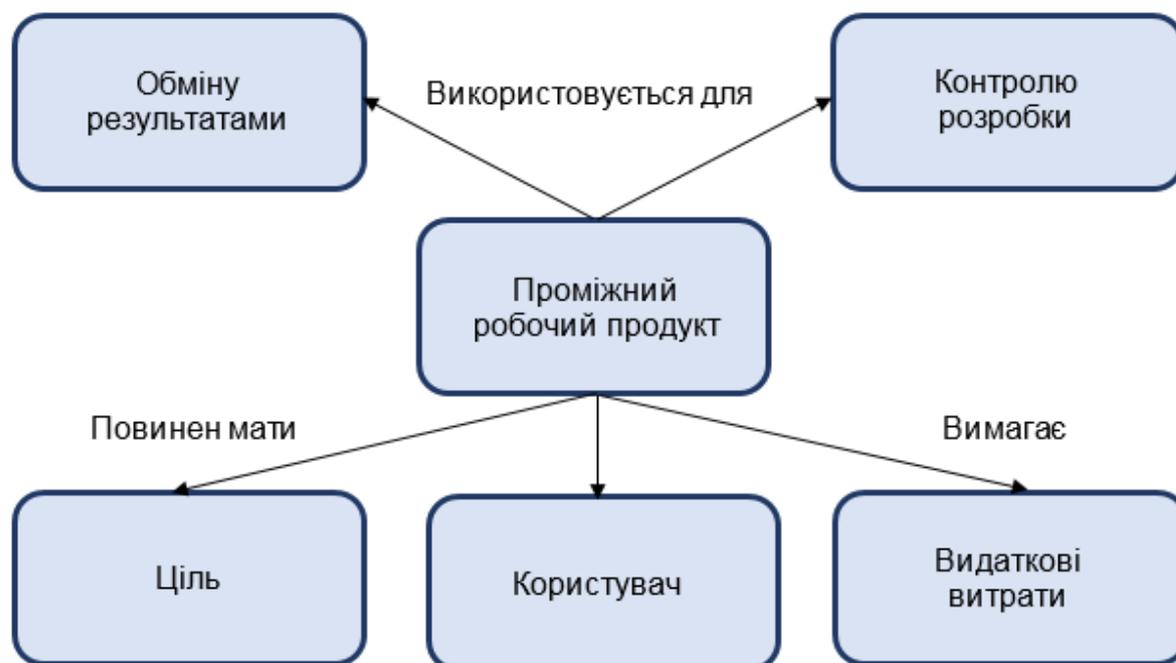


Рис. 3.2 Атрибути проміжного робочого продукту

Дисципліна зобов'язань

В основі розподілу обов'язків в бізнесі і в промисловому виробництві, формування корпоративних правил і норм лежить певна ділова етика або форма відносин – дисципліна зобов'язань. Вона широко використовується на практиці і є однією з можливих форм соціального взаємовідношення між людьми. Привнесення в бізнес і промисловість інших моделей людських відносин – сімейних, сексуальних, дружніх і так далі, часто наносить справам серйозні збитки, породжує конфліктність, знижує ефективність.

Основою такої форми відносин, як дисципліна зобов'язань, є зобов'язання, які:

- даються добровільно;
- не даються легко – робота, ресурси, розклад повинні бути ретельно враховані;
- між сторонами включає те, що буде зроблене, ким і в які терміни;
- відкрито і публічно сформульовані (тобто це не "таємне знання").

Крім того:

- відповідальна сторона прагне виконати зобов'язання, навіть якщо потрібна допомога;
- до настання *deadline*, як лише стає очевидно, що робота не може бути закінчена у визначений термін, обговорюються нові зобов'язання.

Відзначимо, що дисципліна зобов'язань не є якимсь зведенням правил і законів, вона також відрізняється і від корпоративної культури. Це – певний груповий психічний феномен, що існує в товаристві сучасних людей. Наведені вище пункти не є вичерпним описом цього феномену, але лише проявляють і позначають його, так би мовити, викликають потрібні спогади.

Дисципліна зобов'язань, хоч і здається цілком очевидною, деколи, не просто реалізується на практиці, наприклад, в творчих галузях людської діяльності, в галузі навчання і так далі. Існують окремі люди, яким ця дисципліна внутрішньо є неприйнятною незалежно від роду їх діяльності.

З іншого боку, люди, які освоїли цю дисципліну, часто прагнуть застосовувати її в інших галузях життя і людських відносин, що виявляється не завжди виправданим. Підкреслимо, що дана дисципліна є далеко не єдиною моделлю відносин між людьми. Як приклад, можна розглянути відносини в сім'ї або дружбу, що, з очевидністю, не можуть бути виражені дисципліною зобов'язань. Так, замість точності і пунктуальності в цих відносинах важливим є емоційно-психологічне співпереживання, без якого вони неможливі.

Дисципліні зобов'язань приділяється багато уваги в межах **MSF** (*Microsoft Solutions Framework (MSF)*— методологія розробки програмного забезпечення, запропонована корпорацією Microsoft. MSF спирається на практичний досвід Microsoft і описує управління людьми і робочими процесами в процесі розробки рішення. MSF є узгоджений набір концепцій, моделей і правил), оскільки там в моделі команди немає лідера, начальника. Ця дисципліна реалізована також в Scrum (підхід управління проектами для гнучкої розробки програмного забезпечення. Скрам чітко робить акцент на якісному контролі процесу розробки): Scrum-команда має багато свобод, і через це – велику відповідальність. Регламентуються також правила дій, коли зобов'язання не можуть бути виконані такою командою.

Проект

Класичний операційний розподіл праці, що йде ще від Адама Сміта, (**Адам Сміт** (англ. Adam Smith; *5 червня 1723 Керкколд, Шотландія — 17 липня 1790 Единбург, Шотландія) — шотландський економіст і філософ-етик; засновник сучасної економічної теорії. Головна праця — «Дослідження про природу і причини багатства народів» (англ. **An Inquiry into the Nature and Causes of the Wealth of Nations**), видана 1776 року. У цій праці Сміт з високою деталізацією систематизує

сучасні йому ідеї. «Поділ праці повинен сприяти та розвивати продуктивність робочої сили найкраще», так розпочинає Сміт свою працю об'ємом понад 1000 сторінок. Особлива увага приділяється розгляду **Переваги поділу праці**. Причиною багатства окремого народу автор вважає розповсюдження поділу праці. Отже, на відміну від фізіократів, які вважали продуктивність властивістю лише землі/природи, саме працю Сміт вважає тим, що забезпечує добробут народу, оскільки забезпечення товарами та послугами стало кращим, проте, якість землі та клімат не змінилися протягом довгого часу. Такий феномен пояснює економічне зростання (як згодом і модель Солоу). Кращими стають уміння, розуміння під час здійснення певного виду роботи. Це, в свою чергу, уможливило саме поділ праці. Він наводить приклад з голками, згідно з яким один працівник може максимально в день виготовляти 20 голок, а 10 працівників, кожний з яких спеціалізується на певному відрізку праці — 48 000 за той самий проміжок часу. Переваги поділу праці можна реалізувати лише в обмінній економіці і настільки ефективно, наскільки функціонує ринок. Сміт резюмує: чим більший розмір ринку, тим більш розвинутим є поділ праці.) і є суттю масового індустріального виробництва. Тобто, існує чітко налагоджений процес роботи і є галузі спеціалізації – один цех точить, інший калібрує, третій збирає, четвертий фарбує і так далі. Продуктивна спроможність такого виробництва набагато перевершує виконання всієї роботи однією людиною або однією групою. Таким чином в XIX столітті операційний розподіл праці став основою мануфактури, що витіснила індивідуальне, ремісничє виробництво. На початку XX століття цю структуру робіт перенесли і на управління – тобто численні менеджери контролювали окремі ділянки робіт.

Проте, високий рівень складності ряду завдань в промисловості і бізнесі не дозволяє (на щастя!) так працювати усюди. Існує багато творчих, нових завдань, де, мабуть, в майбутньому і з'явиться можливість створити конвеєри, але в даний момент для їх вирішення потрібна істотна концентрація сил і *енергії* людей, несподівані рішення, а також успіх і легка рука. Це і є галузь проектів.

Проект – це унікальна (на відміну від традиційного після операційного промислового виробництва) діяльність, що має:

початок і кінець у часі,

спрямована на досягнення певного результату/цілі, створення певного унікального продукту або послуги,

задані обмеження на ресурси і часові терміни,

чіткі вимоги до якості і допустимого рівня ризику.

Зокрема, розробка програмного забезпечення, є, переважно, проектною галуззю.

Необхідно розрізняти проекти промислові і проекти творчі. У них різні принципи управління. Складність промислових проектів – у великій кількості різних організацій, компаній і відносної унікальності самих робіт. Приклад – будівництво багатоповерхового будинку. Сюди ж відносяться

різні міжнародні проекти і не лише промислові – освітні, культурні і ін. Завдання в управлінні такими проектами – це все охопити, все проконтролювати, нічого не забути, все звести воедино, добитися руху, причому руху узгодженого.

Творчі проекти характеризуються абсолютною новизною ідеї – новий сервіс, абсолютно новий програмний продукт, якого ще не було на ринку, проекти в галузі мистецтва і науки. Будь-який початкуючий бізнес, як правило, є таким ось творчим проектом. Причому новизна в подібних проектах не лише абсолютна – такого ще не було. Таке, може, вже і було, але лише не з нами, командою проекту. Тобто присутній величезний об'єм відносної новизни для самих людей, які втілюють цей проект.

Проекти з розробки програмного забезпечення знаходяться між двома цими полюсами і займають в цьому просторі різне положення. Часто вони складні, тому що об'ємні, і знаходяться на стику різних дисциплін – того цільового бізнесу, куди повинен вбудуватися програмний продукт, і складного, нетривіального програмування. Часто сюди додається ще розробка унікального електронно-механічного устаткування. З іншого боку, оскільки програмування активно просувається в різні сфери людської діяльності, то відбувається це шляхом створення абсолютно нових, унікальних продуктів, і їх розробка і просування володіють всіма рисами творчих проектів.

Управління проектами (*project management*) – галузь діяльності, в ході якої, в межах певних проектів, визначаються і досягаються чіткі цілі при знаходженні компромісу між об'ємом робіт, ресурсами (такими як гроші, праця, матеріали, енергія, простір і ін.), часом, якістю і ризиками.

Відзначимо декілька важливих аспектів управління проектами.

- **Stakeholders** – люди «з боку», які не беруть участь безпосередньо в проекті, але впливають на нього і/або зацікавлені в його результатах. Це можуть бути майбутні користувачі системи (наприклад, за ситуації, коли вони і замовник – це не одне і те ж), вище керівництво компанії-розробника і так далі. Ідентифікація всіх stakeholders і виважена робота з ними – важлива складова успішного проектного менеджменту.
- **Project scope** – межі проекту. Це дуже важливе поняття для програмних проектів з причини мінливості вимог. Часто буває, що розробники починають створювати одну систему, а потім, поступово, вона перетворюється на іншу. Причому для менеджерів з продаж, а також замовника, нічого радикально не відбулося, а з погляду внутрішньої структури ПЗ, технологій, алгоритмів реалізації, архітектури – все радикально змінюється. За подібними тенденціями повинен

стежити і ретельно з ними розбиратися проектний менеджмент.

- **Компроміси – найважливіший аспект управління програмними проектами в силу погодження ПЗ. Важливо не втратити всі параметри і сторони, що погоджуються, та знайти прийнятний компроміс.**

Під час розробки програмних проектів, виходячи з досвіду і пропозицій **MSF 3.1**, важливими є такі галузі управління.

(Модель процесів MSF - Microsoft Solutions Framework - методологія розробки програмного забезпечення, запропонована корпорацією Microsoft. MSF спирається на практичний досвід Microsoft і описує управління людьми і робочими процесами в процесі розробки рішення. (MSF process model) представляє загальну методологію розробки та впровадження ІТ рішень. Особливість цієї моделі полягає в тому, що завдяки своїй гнучкості та відсутності жорстко нав'язуваних процедур вона може бути застосована при розробці досить широкого кола ІТ проектів. Ця модель поєднує в собі властивості двох стандартних виробничих моделей: каскадної (waterfall) і спіральної (spiral). Модель процесів в MSF 3.0 була доповнена ще одним інноваційним аспектом: вона покриває весь життєвий цикл створення рішення, починаючи з його відправної точки і закінчуючи безпосередньо впровадженням. Такий підхід допомагає проектним групам сфокусувати свою увагу на бізнес-вартості (business value) рішення, оскільки ця вартість стає реальною лише після завершення впровадження та початку використання продукту.

Процес MSF орієнтований на «віхи» (mile stones) - ключові точки проекту, що характеризують досягнення в його рамках будь-якого істотного (проміжного або кінцевого) результату. Цей результат може бути оцінений і проаналізований, що має на увазі відповіді на питання: «Прийшла чи проектна група до однозначного розуміння цілей і рамок проекту?», «Чи достатньою мірою готовий план дій?», «Чи відповідає продукт затвердженій специфікації?», « чи задовольняє рішення потреби замовника? » і т. д.

Модель процесів MSF враховує постійні зміни проектних вимог. Вона виходить з того, що розробка рішення повинна складатися з коротких циклів, що створюють поступальний рух від найпростіших версій рішення до його остаточного вигляду.

Модель процесів MSF тісно пов'язана з базовими принципами MSF, розглянутими вище. Взагалі кажучи, трьома особливостями моделі процесів MSF є:

- Підхід, заснований на фазах і віхи.
- Ітеративний підхід.
- Інтегрований підхід до створення та запровадження рішень.

Модель процесів включає такі основні фази процесу розробки:

- Вироблення концепції (Envisioning)
- Планування (Planning)
- Розробка (Developing)
- Стабілізація (Stabilizing)
- Впровадження (Deploying)

<u>Галузь управління проектами</u>	<u>Опис</u>
<u>Планування і моніторинг проекту, контроль за змінами в проекті (Project planning/tracking/change Control).</u>	<u>Інтеграція і синхронізація планів проекту; організація процедур і систем управління і моніторингу проектних змін</u>
<u>Управління межами проекту (Scope Management).</u>	<u>Визначення і розподіл об'єму робіт (меж проекту); управління компромісними рішеннями в проекті</u>
<u>Управління календарним графіком проекту (Schedule Management).</u>	<u>Складання календарного графіку виходячи з оцінок трудовитрат, впорядкування завдань, співвідношення доступних ресурсів із завданнями, застосування статистичних методів, підтримка календарного графіку</u>
<u>Управління вартістю (Cost management).</u>	<u>Оцінка вартості виходячи з оцінок тимчасових витрат; звітність про перебіг проекту і його аналіз; аналіз витратних ризиків; функціонально-вартісний аналіз (value analysis)</u>
<u>Управління персоналом (Staff Resource Management).</u>	<u>Планування ресурсів; формування проектної команди; вирішення конфліктів; планування і управління підготовкою</u>
<u>Управління комунікаціями (Communications Management)</u>	<u>Комунікаційне планування (між проектною групою, замовником/спонсором, споживачами/користувачами, ін. зацікавленими особами); звітність про перебіг проекту</u>
<u>Управління ризиками (Risk management).</u>	<u>Організація процесу управління ризиками в команді і сприяння йому; забезпечення документообігу управління ризиками</u>
<u>Управління постачанням (Procurement).</u>	<u>Аналіз цін постачальників послуг і/або апаратного/програмного забезпечення; підготовка документів про ініціацію пропозицій (requests for proposals – RFPS), вибір постачальників і субпідрядників; складання контрактів і переговори про їх умови; договори; замовлення на постачання і платіжні вимоги</u>
<u>Управління якістю (Quality management).</u>	<u>Планування якості, визначення вживаних стандартів, документування критеріїв якості і процесів його вимірювання</u>