

УДК 004.415/416

DOI: 10.31673/2412-9070.2024.020308

І. М. ГАМАНЮК, ст. викладач;

І. М. АВЕРІЧЕВ, канд. екон. наук, доцент,

Державний університет інформаційно-комунікаційних технологій, Київ

МАНІФЕСТ ГНУЧКОЇ РОЗРОБКИ AGILE З ПОГЛЯДУ ЗАМОВНИКА

На цей час одним із найпопулярніших підходів до розроблення програмного продукту є підхід Agile. Він визначається маніфестом гнучкої розробки. Значну роль у досягненні успіху в проєкті відіграють процеси діяльності замовника. Постає запитання: «А який вигляд матиме маніфест Agile на думку замовника?». Запропоновано розглянути маніфест гнучкої розробки Agile з погляду замовника з метою відпрацювання вимог до замовника. Визначено важливі напрями для замовника, щоб забезпечити успіх у проєкті.

Ключові слова: маніфест гнучкої розробки Agile; розроблення програмного забезпечення; замовник розроблення програмного продукту; розробник програмного продукту; тестування; дослідна експлуатація.

Вступ

У процесі створення програмного продукту беруть участь зазвичай дві сторони: замовник і розробник. Проблема полягає в тому, що замовник бажає отримати певну користь від упровадження тієї чи тієї інновації. Але погляди замовника на програмний продукт, зокрема неправильне визначення місця і ролі програмного продукту у своїй системі та некомпетентні дії замовника під час взаємодії з розробником, можуть призвести до значного зменшення ефекту від упровадження нового програмного продукту. Розробник бажає мати певний економічний ефект від отримання замовлення на розроблення програмного продукту, і тому він не зацікавлений у діях, спрямованих на відмову замовника від упровадження нового програмного продукту. Зазвичай розробник не несе відповідальності за отримання чи неотримання користі від упровадження замовником нового програмного продукту. Усю відповідальність покладено на замовника.

Вирішення проблем щодо визначення місця і ролі програмного продукту в системі замовника лежить у площині науки бізнес-аналізу.

Стосовно дій із розроблення програмного продукту написано багато. Здебільшого це викладено з погляду розробника. А щодо дій із розроблення програмного продукту з міркування замовника написано мало. Так, у статті [5] визначено, що участь користувачів має бути спрямована на забезпечення розробників потрібними знаннями з предметної галузі, а також має розглядатися як один із засобів у досягненні успішності в проєктах із розроблення інформаційних систем. У праці [4] зазначено, що користувач — це фахівець у своїй галузі, він знає чого хоче. Але крім професійних знань у предметній галузі, користувач має володіти знаннями у сфері інформаційних технологій для правильної постановки завдання. Також тут розглянуто План постановки завдання замовником з автоматизації оброблення інформації, зо-

крема сутність задачі, опис вхідної і вихідної інформації, опис алгоритму розв'язання задачі та опис умовно-постійної інформації.

Мета статті — вирішення проблем взаємодії замовника з розробником, зокрема співпраці під час використання Agile-підходу в процесі розроблення програмного продукту.

Основна частина

Agile — це підхід у розробленні програмного забезпечення. Він визначається маніфестом гнучкої розробки, який було створено й ухвалено в лютому 2001 року. Agile не розглядає практик, а базується на чотирьох цінностях та на 12-ти принципах, якими керуються успішні команди.

Оскільки значну частку в процесі досягнення успіху в проєкті перебирає на себе діяльність замовника, то постає запитання: «А який вигляд матиме маніфест Agile з погляду замовника?».

А отже, щоб відповісти на це запитання, потрібно більш ґрунтовно розкрити сутність ідей і принципів маніфесту Agile.

1. Основні ідеї (цінності) маніфесту Agile [1–3].

1.1. Особистості та їхні взаємодії важливіші, ніж процеси та інструменти.

1.2. Робоче програмне забезпечення важливіше, ніж повна документація.

1.3. Співпраця із замовником важливіша, ніж контрактні зобов'язання.

1.4. Реакція на зміни важливіша, ніж дотримання плану.

2. Принципи, які роз'яснює маніфест Agile.

2.1. Задоволення клієнта завдяки ранньому та безперервному постачанню вартісного програмного забезпечення.

2.2. Схвальне ставлення до змін щодо вимог навіть наприкінці розроблення (це може підвищити конкурентоспроможність отриманого продукту).

2.3. Багаторазове постачання робочого програмного забезпечення (кожен місяць або тиждень, або ще частіше).

2.4. Тісне, щоденне спілкування замовника з розробниками впродовж усього проєкту.

2.5. Проєктом опікуються вмотивовані особистості, забезпечені належними умовами роботи, підтримкою і довірою.

2.6. Рекомендований метод передавання інформації — особиста розмова (віч-на-віч).

2.7. Робоче програмне забезпечення — найкращий вимірник прогресу.

2.8. Спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений строк.

2.9. Повсякчасна увага поліпшенню технічної досконалості та зручному дизайну.

2.10. Простота — мистецтво не робити зайвої роботи.

2.11. Найкращі технічні вимоги, дизайн та архітектура притаманні лише самоорганізованій команді.

2.12. Постійна адаптація до мінливих обставин.

Розглянемо докладніше основні ідеї маніфесту.

1.1. Особистості та їхні взаємодії важливіші, ніж процеси та інструменти.

Ця ідея спонукає замовника:

- відряджати особистості для взаємодії з розробником;

- беручи до уваги те, що взаємодії особистостей важливіші, ніж процеси та інструменти, замовник має так чи інакше брати участь у процесах розроблення програмного забезпечення та використанні інструментів розроблення;

- для того, щоб підвищити якість взаємодії особистостей, потрібно добирати та підготовляти особистості до взаємодії;

- здійснювати підготовку особистостей за напрямом бізнес-правил діяльності замовника та діяльності розробника. Замовник повинен підготувати своїх особистостей до процесу створення програмного забезпечення та до використання інструментів розроблення;

- продумати підвищення якості взаємодії через створення та сприяння поліпшенню середовища взаємодії;

- виділити час та відповідні ресурси;

- у разі обмеження ресурсів віддавати перевагу особистостям і їх взаємодії з особистостями розробника, ніж процесам та інструментам розроблення.

1.2. Робоче програмне забезпечення важливіше, ніж повна документація.

Ця ідея спонукає замовника:

- чітко розуміти, що являє собою робоче програмне забезпечення та повна документація;

- мати відповідний рівень підготовки особистостей для того, щоб відрізнити робоче програмне забезпечення від неробочого;

- мати відповідні ресурси та документацію для здійснення процесу визначення робочого програмного забезпечення;

- з огляду на те, що робоче програмне забезпечення важливіше, ніж повна документація, замовник має так чи інакше брати участь у процесах розроблення повної документації. Особливо у прийнятті елементів документації, відпрацьованих після чергового етапу;

- для підвищення якості прийняття елементів документації особистості потрібно належно добирати і готувати їх до цього процесу;

- підготовку особистостей здійснювати за напрямом бізнес-правил діяльності замовника та діяльності розробника. Замовник має підготувати своїх особистостей до процесу створення робочого програмного забезпечення та до процесу розроблення повної документації;

- ретельно продумувати підвищення якості взаємодії через створення та сприяння подальшому поліпшенню середовища взаємодії щодо окреслення робочого програмного забезпечення та прийняття елементів документації;

- виділяти відповідні ресурси на визначення робочого програмного забезпечення та прийняття елементів документації;

- у разі обмеження ресурсів (наприклад, часових) віддавати перевагу робочому програмному забезпеченню, ніж прийняттю повної документації.

1.3. Співпраця із замовником важливіша, ніж контрактні зобов'язання.

Ця ідея спонукає замовника:

- співпрацювати з розробником;

- беручи до уваги те, що співпраця із замовником важливіша, ніж контрактні зобов'язання, замовник повинен так чи інакше мати контрактні зобов'язання та проводити щодо них переговори;

- підготовку особистостей здійснювати за напрямом організації якісної співпраці, укладання контрактних зобов'язань та ведення переговорів стосовно них;

- продумувати подальше підвищення якості взаємодії через створення та сприяння поліпшенню середовища співпраці;

- виділяти відповідні ресурси на співпрацю із розробником, розроблення контрактних зобов'язань та проведення щодо них переговорів;

- у разі обмеження ресурсів віддавати перевагу співпраці із розробником, ніж обговоренню контрактних зобов'язань.

1.4. Реакція на зміни важливіша, ніж дотримання плану.

Ця ідея спонукає замовника:

- вносити пропозиції щодо змін програмного забезпечення, які необхідно здійснити;

- мати достатній рівень тестування програмного забезпечення для виявлення відхилень від бажаного продукту;

- мати відповідний рівень підготовки особистостей для тестування програмного забезпечення та підготовки пропозицій до змін програмного забезпечення;

- мати план та дотримуватися його;
- бути готовими, що план може бути не виконано вчасно через внесені пропозиції та їх реалізацію;
- виділяти відповідні ресурси на тестування програмного забезпечення, розроблення та дотримання плану;
- у разі обмеження ресурсів віддавати перевагу внесенню пропозицій, ніж дотриманню плану.

Далі більш детально розглянемо принципи, які роз'яснює маніфест Agile.

2.1. Задоволення клієнта завдяки ранньому та безперервному постачанню вартісного програмного забезпечення.

Цей принцип спонукає замовника:

- мати підготовлених фахівців із тестування та експлуатації дослідного зразка;
- брати участь у тестуванні продукту до встановлення на дослідну експлуатацію, щоб бути впевненим, що дослідний зразок відповідає вимогам для дослідної експлуатації. Ці вимоги мають бути визначені заздалегідь;
- бути готовим до отримання ранньої, ще недоопрацьованої системи, або частини системи і встановлення її на дослідну експлуатацію;
- слід застосовувати не всі засоби для роботи з дослідним зразком, а лише ту частину, якої буде достатньо для тестування зразка та надання пропозицій. Можливо здійснити пілотний проект на одній із ділянок;
- потрібно продумувати питання щодо дій у разі виникнення проблем у недоопрацьованому зразку;
- бути готовим до появи певних перебоїв у експлуатації дослідного зразка;
- спланувати та виділяти відповідні ресурси на дослідну експлуатацію;
- бути готовим для оцінювання як позитивних, так і негативних наслідків дослідної експлуатації. Мати чітку систему вимірювань.

2.2. Схвальне ставлення до змін щодо вимог навіть наприкінці розроблення (це може підвищити конкурентоспроможність отриманого продукту).

Цей принцип спонукає замовника:

- генерувати зауваження та пропозиції щодо продукту на всіх етапах розроблення продукту, навіть наприкінці розроблення;
- бути готовим до того, що не всі зауваження та пропозиції будуть реалізовані;
- бути готовим до того, що можуть статися збої в роботі продукту, або відступ назад на попередні позиції розроблення через втрату часу на усунення зауважень або реалізацію пропозицій;

- бути готовим до того, що може змінитися вся концепція продукту, що розробляється.

2.3. Багаторазове поставляння робочого програмного забезпечення (кожен місяць або тиждень, або ще частіше).

Цей принцип спонукає замовника:

- вносити зміни до бізнес-правил, щоб уникнути збоїв у роботі під час отримання нової версії робочого програмного забезпечення;
- брати до уваги, що з кожною новою версією може бути поставлено нову функціональність, призводячи до залучення до дослідної експлуатації щоразу більшої кількості працівників;
- затримувати строки виконання проекту, спричинені обмеженнями щодо отримання нових версій продукту;
- вносити зміни до бізнес-правил, щоб найшвидше охопити тестовим покриттям увесь продукт, що тестується, та надати відгуки щодо продукту.

2.4. Тісне, щоденне спілкування замовника з розробниками впродовж усього проекту.

Цей принцип спонукає замовника:

- до створення умов для здійснення тісного щоденного спілкування своїх представників із розробником упродовж усього проекту, а саме: підготовки своїх представників, виділення ресурсів (фінансових, людських, часових тощо);
- створення системи опрацювання звітів своїх представників та своєчасного реагування на появу відповідної інформації;
- після закінчення проекту бути готовим до переходу найбільш кваліфікованих представників замовника до розробника для участі як бізнес-аналітиків у подальших проектах розробника.

2.5. Проектом опікуються вмотивовані особистості, забезпечені належними умовами роботи, підтримкою і довірою.

Цей принцип спонукає замовника:

- розуміти важливість відповідної оплати діяльності розробника та сприяти мотивації особистостей замовника, залучених до участі у проекті;
- цікавитися так чи інакше внутрішніми проблемами розробника та сприяти їх вирішенню, щоб це не завадило досягненню успіху проекту;
- сприяти створенню комфортних умов для роботи.

2.6. Рекомендований метод передавання інформації — особиста розмова (віч-на-віч).

Цей принцип спонукає замовника:

- сприяти створенню комфортних умов для роботи, зокрема у формі особистих розмов (віч-на-віч).

2.7. Робоче програмне забезпечення — найкращий вимірник прогресу.

Цей принцип спонукає замовника:

- створити чітку систему зворотнього зв'язку з відстеження та реагування на виклики, в якій

основною шкалою вимірювання прогресу є робоче програмне забезпечення;

- чітко розуміти, а за потреби фіксувати в документах, що є робочим програмним забезпеченням;
- мати підготовлених фахівців із тестування та вимірювання прогресу.

2.8. Спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений строк.

Цей принцип спонукає замовника:

- відслідковувати процес створення продукту та здійснювати вплив на підтримання постійного темпу з чітким розумінням: що створено, що планується створити, чи є достатньо ресурсів для підтримання постійного темпу роботи;
- виявляти прогалини у своїй роботі, що можуть призвести до затримання та відхилення від графіка робіт (обсягу виконаних робіт).

2.9. Повсякчасна увага поліпшенню технічної досконалості та зручному дизайну.

Цей принцип спонукає замовника:

- вносити вимоги, що поліпшують технічну досконалість продукту;
- вносити вимоги, що поліпшують зручність дизайну продукту;
- необхідно мати підготовлених фахівців із технічної досконалості та дизайну продукту;
- сприяти модульному підходу до створення архітектури в разі повторного використання модулів у подальшому в проекті, що зі свого боку підвищить швидкість створення продукту та сприятиме концентруванню ресурсів та зусиль на інших важливих напрямках.

2.10. Простота — мистецтво не робити зайвої роботи.

Цей принцип спонукає замовника:

- ранжувати вимоги в такий спосіб, щоб швидко отримувати від розробника прості рішення, які б давали певну користь і могли б бути базовими для подальшої роботи;
- відслідковувати, щоб простота дійсно зумовлювала зменшення зайвої роботи;
- мати підготовлених фахівців, які б змогли аналізувати роботу розробника.

2.11. Найкращі технічні вимоги, дизайн та архітектура притаманні лише самоорганізованій команді.

Цей принцип спонукає замовника:

- відслідковувати процес самоорганізації команди розробника;
- оцінювати результат роботи розробника за фаховим напрямом членів команди;
- мати фахівців з організації командної роботи.

2.12. Постійна адаптація до мінливих обставин.

Через рівні проміжки часу команда розмірковує над тим, щоб стати більш ефективною, потім налаштовує свою поведінку.

Цей принцип спонукає замовника:

- мати відповідні ресурси для здійснення адаптації до мінливих обставин;
- сприяти своїй команді та команді розробника зрозуміти, що потрібно для того, щоб стати більш ефективними;
- сприяти своїй команді та команді розробника налаштувати свою поведінку.

Висновки

Отже, можна дійти таких висновків, що саме є важливим з боку замовника.

1. У сфері підготовки фахівців:

- здійснювати підготовку фахівців у галузі діяльності замовника, розроблення програмного продукту, тестування та вимірювання прогресу, технічної досконалості та дизайну продукту, аналізу роботи розробника, організації командної роботи, якості документації до продукту, організації якісної співпраці, підготовки контрактних зобов'язань та веденню переговорів стосовно них, тестування дослідного зразка для виявлення відхилень від бажаного продукту, для усунення можливостей отримання незадовільних результатів, тестування та експлуатації дослідного зразка;

- чітко розуміти, що являє собою робоче програмне забезпечення та повна документація. Мати відповідний рівень підготовки особистостей для того, щоб відрізнити робоче програмне забезпечення від неробочого (рівень працездатності).

2. У сфері організації взаємодії:

- підвищувати якість взаємодії завдяки створенню та сприянню поліпшенню середовища взаємодії, сприяти створенню комфортних умов для роботи, зокрема у формі особистих розмов (віч-на-віч), прийняття елементів документації, прийняття продукту;
- віддавати перевагу особистостям і їх взаємодії з особистостями розробника, ніж процесам та інструментам розроблення.

3. У сфері виділення ресурсів:

- потрібно виділяти відповідні ресурси (часові, людські, фінансові тощо);
- мати відповідні ресурси для здійснення адаптації до мінливих обставин, на реагування щодо змін (вимог до продукту, самого продукту, планів тощо), на прийняття елементів документації, дослідну експлуатацію та ризики.

4. У сфері створення продукту:

- генерувати та обговорювати зауваження та пропозиції щодо продукту на всіх етапах розроблення продукту, навіть наприкінці розроблення. Бути готовим до того, що може змінитися концепція продукту, що розробляється;
- мати системи опрацювання звітів своїх представників та своєчасного реагування на появу відповідної інформації;

- мати чітку систему зворотного зв'язку з відслідковування та реагування на виклики, в якій основною шкалою вимірювання прогресу є робочий продукт;

- чітко розуміти та за потреби фіксувати в документах, що є робочим продуктом;

- відслідковувати процес створення продукту та здійснювати вплив на підтримання постійного темпу з чітким розумінням, що створено, що планується створити, чи є достатньо ресурсів для підтримання постійного темпу роботи;

- виявляти прогалини у своїй роботі, що можуть призвести до затримання та відхилення від графіку робіт (обсягу виконаних робіт);

- вносити вимоги, що поліпшують технічну досконалість продукту, зручність дизайну продукту та щодо змін у продукті, які потрібно здійснити;

- сприяти модульному підходу до створення архітектури під час повторного використання модулів у подальшому в проекті, що зі свого боку підвищить швидкість створення продукту та сприятиме концентрації ресурсів та зусиль на інших важливих напрямках;

- ранжувати вимоги в такий спосіб, щоб отримувати швидко від розробника прості рішення, які б приносили певну користь і які б могли бути базовими для подальшої роботи;

- мати відповідні ресурси та відповідні документи для здійснення процесу визначення робочого програмного забезпечення, чи відповідає документація відповідним вимогам;

- віддавати перевагу робочому програмному забезпеченню, ніж прийняттю повної документації;

- мати контрактні зобов'язання, але в разі обмеження ресурсів та отримання незначної шкоди для замовника в результаті неякісного виконання розробником своїх контрактних зобов'язань, віддавати перевагу співпраці із розробником, ніж обговоренню контрактних зобов'язань;

- мати план та дотримуватися виконання плану. Бути готовими, що план може бути не виконано вчасно через внесені пропозиції та їх реалізацію;

- ухвалювати зважені рішення та намагатись віддавати перевагу внесенням змін, ніж дотриманню планів.

5. У сфері дослідної експлуатації:

- визначати попередньо вимоги до продукту за рейтингом (що необхідно в першу чергу, другу чергу тощо);

- бути готовим до отримання раннього, ще недоопрацьованого продукту, або частини продукту і встановлення його на дослідну експлуатацію. Слід застосовувати не всі засоби для роботи з дослідним зразком, а тільки ту частину, якої буде достатньо для тестування зразка та надання пропозицій. Розглядати можливість здійснення пілотного проекту на одній із ділянок;

- продумати питання щодо дій (можливо тимчасовий відхід від використання продукту) у разі виникнення проблем у недоопрацьованому зразку;

- бути готовим для оцінювання як позитивних, так і негативних наслідків дослідної експлуатації;

- мати чітку систему вимірювань;

- вносити зміни до бізнес-правил замовника, щоб уникнути збоїв у роботі під час отримання нової версії робочого продукту. Оскільки з кожною новою версією може бути поставлено нову функціональність, то це може спричинити залучення до дослідної експлуатації щоразу більшої кількості працівників. Обмеження щодо отримання нових версій продукту може призвести до затримання строків виконання проекту;

- вносити зміни до бізнес-правил, щоб найшвидше охопити тестовим покриттям увесь продукт, що тестується, та надати відгуки щодо продукту.

6. У сфері мотивації фахівців:

- розуміти важливість відповідної оплати діяльності розробника та сприяти мотивації особистостей замовника, залучених до участі у проекті;

- після закінчення проекту бути готовими до переходу найбільш кваліфікованих представників замовника до інших замовників, або до розробників для участі в наступних проектах як бізнес-аналітиків.

7. У сфері опрацювання ризиків:

- цікавитися так чи інакше внутрішніми проблемами розробника та сприяти їх вирішенню, щоб це не завадило досягненню успіху проекту;

- ознайомлюватися з процесом самоорганізації команди розробника;

- сприяти своїй команді та команді розробника стосовно розуміння, що саме потрібно, щоб стати більш ефективними, та налаштувати свою поведінку.

Список використаної літератури

1. Ларман К. Використання UML 2.0 та шаблонів проектування. Практичне керівництво 3-є видання / пер. з англ. Видавничий дім «Вільямс», 2013. 736 с.

2. Мартін Р., Мартін М. Принципи, патерни і методики гнучкої розробки мовою С# / пер. з англ. Видавництво «Символ-Плюс», 2014. 768 с.

3. Agile маніфест [Електронний ресурс]. URL: <http://www.agileukraine.org/p/agile-manifesto.html>

4. Роль замовника у створенні інформаційної системи [Електронний ресурс]. URL: <http://um.co.ua/4/4-3/4-35937.html>

5. He J., King W. R. The Role of User Participation in Information Systems Development: Implications from a Meta-Analysis // Journal of Management Information Systems, Vol. 25, No. 1 (Summer, 2008). P. 301–331.

I. Gamaniuk, I. Averichev

AGILE FLEXIBLE DEVELOPMENT MANIFESTO FROM THE CUSTOMER'S POINT OF VIEW

Manifesto for Agile Software Development is written from the developer's point of view. In this article it is offered to consider the manifesto from the point of view of the customer. Having elaborated the main ideas and principles of the Agile Manifesto, conclusions are made to each idea and to each principle, they are generalized and presented.

Important on the customer's side are:

In the field of training: to train specialists in the field of customer service, software development, testing and measurement of progress, technical excellence and product design analysis, teamwork, quality of product documentation, quality cooperation, preparation of contractual obligations and negotiations on them, testing of the prototype.

In the field of interaction: to improve the quality of interaction by creating and promoting the improvement of the interaction environment, to promote the creation of comfortable working conditions, including in the form of personal conversations (face to face), the adoption of elements of documentation, product acceptance; give preference to individuals and their interaction with the developer's personalities than to development processes and development tools;

In the field of resource allocation: it is necessary to allocate appropriate resources (time, human, financial and others); have the appropriate resources to adapt to changing circumstances, to respond to changes, to accept elements of documentation, to pilot operation and risks.

In the field of product development: generate and discuss product comments and proposals at all stages of product development, even at the end of development. Be prepared for the fact that the concept of the product being developed may change; creation of a system for processing the reports of its representatives and timely response to the appearance of relevant information; create a clear feedback tracking and response system in which the main scale for measuring progress is the working product; clearly understand and possibly record in documents that are a working product; monitor the product development process and influence the maintenance of a constant pace with a clear understanding of what is created, what is planned to be created, whether there are enough resources to maintain a constant pace of work; identify gaps in their work that can lead to delays and deviations from the work schedule (volume of work performed); make requirements that improve the technical perfection of the product, the convenience of product design and the changes in the product that need to be made; to promote a modular approach to the creation of architecture, with the reuse of modules later in the project, which in turn will increase the speed of product creation and promote the concentration of resources and efforts in other important areas; rank the requirements in such a way as to quickly obtain from the developer simple solutions that would bring some benefits and that could be the basis for further work; have the appropriate resources and documents to carry out the process of determining whether the working software meets the relevant requirements; prefer working software to the adoption of complete documentation; have contractual obligations, but with limited resources and minor damage to the customer as a result of poor performance by the developer of its contractual obligations, to prefer cooperation with the developer than to discuss contractual obligations; have a plan and follow the plan. Be prepared that the plan may not be implemented on time due to proposals and their implementation; make informed decisions and try to prioritize change over adherence to plans;

In the field of experimental operation: determine in advance the requirements for the product by rating (what is needed first, second, etc.); be prepared to receive an early, unprocessed product or part of the product and install it for trial operation. Not all tools should be used to work with the prototype, but only the part that will be sufficient to test the sample and make suggestions. Consider the possibility of implementing a pilot project at one of the sites; think about questions about actions (possibly temporary withdrawal from the use of the product) in case of problems in the unfinished sample; be prepared to assess both the positive and negative effects of experimental operation; have a clear system of measurements; make changes to the customer's business rules to avoid malfunctions when receiving a new version of the working product. Given that new functionality can be added with each new version, this can lead to more and more employees involved in trial operation. Restrictions on obtaining new versions of the product may lead to delays in project implementation; make changes to the business rules to cover the entire test product as quickly as possible and provide product feedback.

In the field of motivation of specialists: understand the importance of appropriate payment for the activities of the developer and to promote the motivation of the customer's personalities involved in the project; at the end of the project to be ready for the transfer of the most qualified representatives of the customer to other customers, or to developers to participate as business analysts in future projects;

In the field of risk processing: be interested to some extent in the internal problems of the developer and contribute to their solution so that it does not interfere with the success of the project; get acquainted with the process of self-organization of the development team; help your team and the development team understand what it takes to become more effective and adjust their behavior.

Keywords: manifesto of flexible agile development; software development; the customer of software product development; software product developer; testing; experimental operation.