



Унифицированный процесс разработки ПО

Управляемый вариантами использования.

Архитектурно-ориентированный.

Итеративный и инкрементный.

ПОНЯТИЯ

Артефакт - это общее название для любых видов *информации*, создаваемой, изменяемой, или используемой сотрудниками при создании системы. Наиболее интересный тип артефактов, используемых в унифицированном процессе разработки ПО - это *модели*.

Модель – это абстракция (знания), описывающая моделируемую систему с определенной *точки зрения* и на определенном уровне *абстрагирования*. Под *точкой зрения* будем понимать представление аналитика требований или проектировщика.

Процесс разработки – набор деятельности, необходимых для переработки *требований заказчика* в согласованный набор *артефактов*, представляющих собой *программное обеспечение*, а позднее для переработки изменений в этих требованиях в новые версии *программного обеспечения*.

Понятие «процесс» в контексте унифицированного процесса (*RUP*), - рассматривается как *шаблон (комплекс знаний)*, который может быть неоднократно использован для создания его экземпляров. Такое понимание сравнимо с пониманием *класса* и *объекта* в объектно-ориентированном проектировании.

Экземпляр процесса – **синоним проекта**.

Задачи унифицированного процесса (вопросы):

1. Как **управлять** деятельностью команды и **проектом** в целом.
2. Какие поставить (определить) **задачи** для отдельного разработчика и команды в целом.
3. Какой перечень **артефактов** следует разработать.
4. Какие необходимы **критерии** для отслеживания и измерения **продуктов** и функционирования проекта.



Унифицированный процесс должен быть: управляемый вариантами использования, архитектурно-ориентированный, итеративный и инкрементный.

Унифицированный процесс – управляемый вариантами использования

Вариант использования – это часть функциональности системы, необходимая для получения пользователем значимого и измеримого результата.

Сумма всех вариантов использования составляет **модель вариантов использования**, которая описывает полную функциональность системы.

Эта модель заменяет традиционное описание функций системы (*функциональной структуры*).

Описание варианта использования отвечает на вопрос, что **система может сделать для каждого пользователя**.



Процесс разработки, **управляемый вариантами использования** означает, что в процессе разработки выполняются серии **рабочих процессов** (отрабатываются управляющие воздействия), **порожденные вариантами использования.**

Поскольку варианты использования управляют процессом разработки, то они разрабатываются в паре с *архитектурой системы*.

Таким образом, варианты использования управляют архитектурой, а архитектура оказывает влияние на варианты использования.

Причем, и варианты использования и архитектура развиваются в процессе *жизненного цикла*.

Унифицированный процесс – ориентирован на архитектуру

Архитектура - это представление всего проекта с выделением ключевых составляющих и затушевывание деталей. *Архитектура* вырастает из требований к результату, в том виде, как их понимает пользователь и другие заинтересованные лица.

Каждый продукт имеет функции и форму, причем одно без другого не существует.

Функции соответствуют *вариантам использования*, а форма – *архитектуре*.

Согласно методологии унифицированного процесса **сначала должны быть разработаны варианты использования**, то есть *функции*, а потом для того чтобы обеспечить выполнение этих функций разрабатывается *архитектура* системы.

С другой стороны архитектура должна обеспечить реализацию необходимых сейчас и в будущем *функций*, то есть *вариантов использования*.

Архитектура и варианты использования разрабатываются параллельно.

Архитектор выполняет следующие работы:

1. Создает грубый набросок архитектуры (эскиз), начиная с той части, которая не связана с вариантами использования (**платформа, ядро и т. п.**). Выделяет *ключевые* варианты использования.
2. Приступает к работе с выделенными ключевыми вариантами использования. Каждый такой вариант использования описывается и реализуется в понятиях **подсистем, классов и компонентов, на основе которых архитектор создает различные модели.**

Архитектура определяется в виде представлений всех **моделей** системы, объединенных (сконфигурированных) в систему.

Существуют архитектурные представления *модели вариантов использования, модели анализа, модели проектирования, модели развертывания.*

Модель реализации включает в себя компоненты, доказывающие то, что архитектура выполнима.

Результатом этой фазы является **базовый уровень архитектуры.**



3. На основе **базового варианта архитектуры**, разрабатывает другие варианты использования.

4. Процесс разработки носит *циклический характер*, так как при разработке очередных вариантов использования архитектору, возможно, потребуется *внести изменения в архитектуру и на базе измененной архитектуры продолжить разработку вариантов использования*.

Процесс разработки продолжается до тех пор, пока архитектура не будет **признана стабильной (удовлетворяющей все требования)**.

Унифицированный процесс – итеративный и инкрементный

Итеративная процедура разработки предполагает наращивание (**инкремент**) функциональности продукта в ходе выполнения стадий разработки.

Для максимальной эффективности итерации должны быть *управляемыми*, то есть они должны быть запланированы и выполняться по плану.

В таком случае, итерации имеют все признаки *проекта*, но поскольку их объемы работ, закладываемые в итерации сравнительно небольшие, то их можно назвать *мини-проектами*.

Задачи, которые образуют итерации, выбираются под воздействием двух факторов:

1. В ходе итерации следует работать с группой вариантов использования, которая *повышает применимость продукта* в ходе дальнейшей разработки.
2. В ходе разработки итерации следует заниматься серьезными *рисками*.



Достоинства управляемого итеративного процесса (в управлении рисками):

1. Управляемая итерация ограничивает *финансовые риски* затратами на одно приращение, так как если разработчикам потребуется повторить итерацию, то затраты будут на одну итерацию, а не стоимость всего продукта.
2. Управляемая итерация снижает *риски не поставки* продукта заказчику в запланированные сроки.
3. Управляемая итерация ускоряет темпы процесса разработки, так как для разработчиков короткий и точный план предпочтительнее длинного и вечно сдвигающегося (*временные риски*).
4. Управляемая итерация признает часто отвергаемый факт, что желания и требования пользователей не могут быть определены в начале разработки (*концептуальные риски*). Они обычно уточняются в последовательных итерациях. Такой подход облегчает адаптацию к изменениям требований.

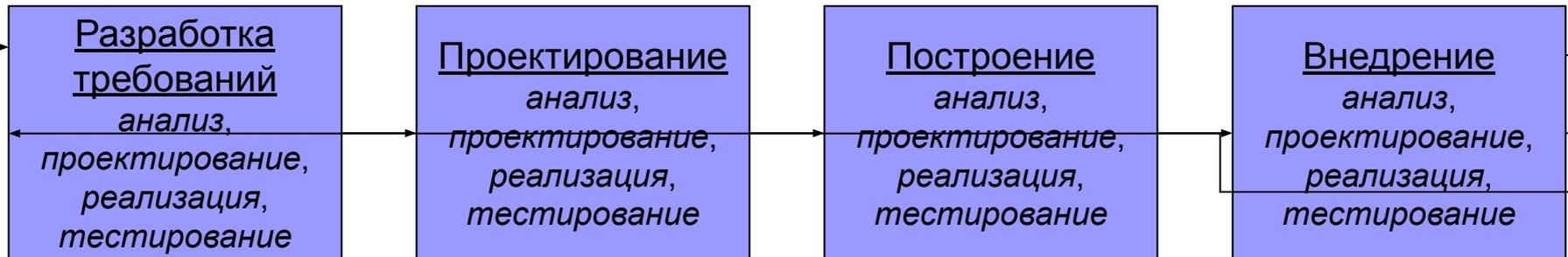
Жизненный цикл в унифицированном процессе

Жизненный цикл информационной системы - последовательность циклов *унифицированного процесса*.

В процессе жизни информационной системы унифицированный процесс циклически повторяется.

Каждый цикл состоит из четырех фаз: (**анализ и разработка требований, проектирование, построение, внедрение**), каждая фаза подразделяется на **итерации**.

Каждая **итерация** включает следующие рабочие процессы: *анализ, проектирование, реализация, тестирование*



I) В ходе фазы **анализа и проектирования требований** идея системы превращается в **концепцию готового продукта** и создается бизнес-план разработки продукта.

В концепции продукта должны быть отражены:

1. Основные *понятия*, которые будет использовать система.
 2. *Функции*, которыми должна обладать система, решая проблемы пользователей.
 3. *План* проекта по разработке системы и *стоимость* разработки.
-

II) В ходе фазы **проектирования** разрабатываются *варианты использования и архитектура системы*.

Архитектура определяется в виде представлений всех **моделей** системы, которые, будучи объединены (сконфигурированы) представляют систему целиком.

Результатом этой фазы является **базовый уровень архитектуры**.



III) В ходе фазы **построения** происходит создание продукта – к архитектуре добавляются законченные программы.

Базовый уровень архитектуры разрастается до продукта, готового к *передаче пользователям*.

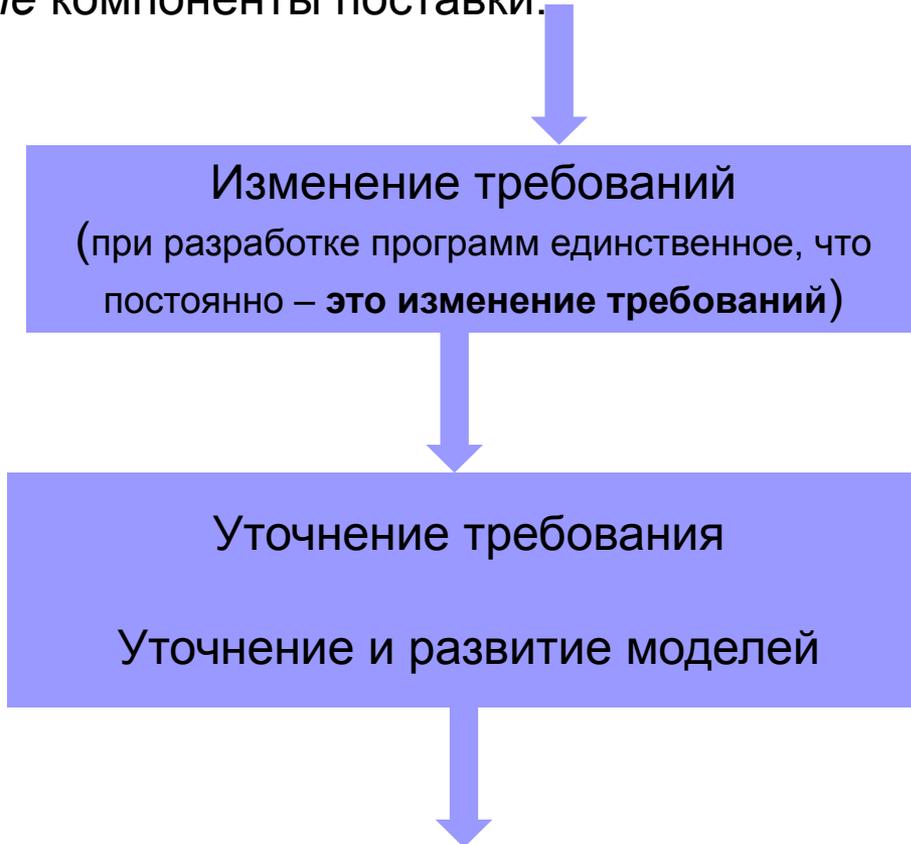
IV) В ходе фазы **внедрения** происходит работа различных пользователей с бета-версией продукта, исправление обнаруженных дефектов, тренинг сотрудников заказчика, поддержка работы пользователей по горячей линии.

Результатом каждого цикла является новый выпуск системы, а каждый выпуск **это продукт готовый к поставке**.

Продукт унифицированного процесса

Готовый к поставке продукт включает:

- а) *исходный код*, воплощенный в компоненты, которые могут быть откомпилированы и проверены;
- б) *руководство пользователя*;
- в) *дополнительные* компоненты поставки.



1. Модель *предметной области* (модель среды окружения).
2. Модель *вариантов использования* системы (функциональная модель).
3. Модель *анализа* (концептуальная модель и первичное распределение поведения).
4. Модель *проектирования*.
5. Модель *реализации*, которая включает в себя компоненты (представленные исходным кодом) и раскладку классов по компонентам.
6. Модель *тестирования*.
7. Модель *развертывания*, которая определяет физические компьютеры-узлы сети и размещение компонентов по этим узлам.

Унифицированный
процесс



Комплекс знаний,
методология разработки