

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

Сегодня развитие программного обеспечения происходит в сторону увеличения и усложнения систем. Мы также желаем получить это программное обеспечение побыстрее. Время выхода на рынок — это другой важный стимул.

Сделать это, однако, нелегко. Наше желание получить мощные и сложные программы не сочетается с тем, как эти программы разрабатываются.

Проблема разработки программного обеспечения сводится к затруднениям разработчиков, вынужденных преодолевать в ходе разработки больших программ множество преград.

Вывод. Общество разработчиков программного обеспечения нуждается в управляемом методе работы. Ему нужен процесс, который объединил бы множество аспектов разработки программ. Ему нужен общий подход, который:

- обеспечивал бы руководство деятельностью команды;
- управлял бы задачами отдельного разработчика и команды в целом;
- указывал бы, какие артефакты следует разработать;
- предоставлял бы критерии для отслеживания и измерения продуктов и функционирования проекта.

Стандартный процесс помогает людям добиваться успеха коллективной работы.

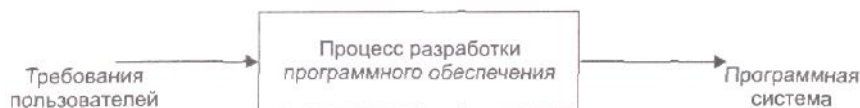
Унифицированный процесс в двух словах

Унифицированный процесс (Rational Unified Process) есть процесс разработки программного обеспечения. Процесс разработки программного обеспечения — это сумма различных видов деятельности, необходимых для преобразования требований пользователей в программную систему (см. рис.).

Унифицированный процесс — это обобщенный каркас процесса, который может быть специализирован для широкого круга программных систем, различных областей применения, уровней компетенции и размеров проекта.

Унифицированный процесс компонентно-ориентирован. Это означает, что создаваемая программная система строится на основе программных **компонентов** связанных хорошо определенными **интерфейсами**.

Специфичные аспекты *Унифицированного процесса* заключаются в трех словосочетаниях — управляемый вариантами использования, архитектурно-ориентированный, итеративный и инкрементный.



Унифицированный процесс управляется вариантами использования

Программная система создается для обслуживания ее пользователей. Для построения успешной системы необходимо знать — в чем нуждаются и чего хотят от системы ее будущие пользователи.

Понятие *пользователь* относится к кому-либо или чему-либо (например, к другой системе, внешней по отношению к рассматриваемой системе), что взаимодействует с системой, которую мы разрабатываем. Пример взаимодействия — человек, использующий банкомат.

Взаимодействие называется **вариантом использования**. *Вариант использования* — это часть функциональности системы, необходимая для получения пользователем значимого для него, ощутимого и измеримого результата. Варианты использования обеспечивают функциональные требования. Сумма всех вариантов использования составляет **модель вариантов использования**, которая описывает полную функциональность системы.

Варианты использования управляют процессом. Они разрабатываются в паре с архитектурой системы.

Унифицированный процесс ориентирован на архитектуру

Понятие архитектуры программы включает в себя наиболее важные статические и динамические аспекты системы.

Требования к результату отражаются в вариантах использования.

Требования к результату также зависят от множества других факторов, таких, как выбор платформы для работы программы, доступность готовых блоков многократного использования, соображения развертывания, унаследованные системы и нефункциональные требования.

Архитектура — это представление всего проекта с выделением важных характеристик и затушевыванием деталей.

Как связаны архитектура и варианты использования? Каждый продукт имеет *функции* и *форму*. Одно без другого не существует. В удачном продукте эти две стороны должны быть уравновешены. Реально архитектура и варианты использования *разрабатываются параллельно*.

Архитектор придает системе *форму*. Архитектура, должна быть спроектирована так, чтобы позволить системе развиваться не только в момент начальной разработки, но и в будущих поколениях.

Унифицированный процесс является итеративным и инкрементным

Разработка коммерческих программных продуктов — это серьезное предприятие, которое может продолжаться от нескольких месяцев до года и более.

Сложную и длительную работу практически было бы разделить на небольшие куски или мини-проекты. Каждый мини-проект является *итерацией*, результатом которой будет приращение. *Итерации* относятся к шагам рабочих процессов, а приращение — к выполнению проекта. Для максимальной эффективности итерации должны быть *управляемыми*, то есть они должны выбираться и выполняться по плану. Поэтому их можно считать мини-проектами.

Управляемый итеративный процесс имеет множество преимуществ.

- Управляемая итерация *ограничивает* финансовые *риски* затратами на одно *приращение*.
- Управляемая итерация *снижает риск непоставки* продукта на рынок в запланированные сроки.
- Управляемая итерация *ускоряет темпы процесса разработки* в целом, *короткий и четкий план* предпочтительнее длинного и вечно сдвигающегося.
- Управляемая итерация признает часто отвергаемый факт — что желания пользователей и связанные с ними требования *не могут быть определены* в начале разработки.

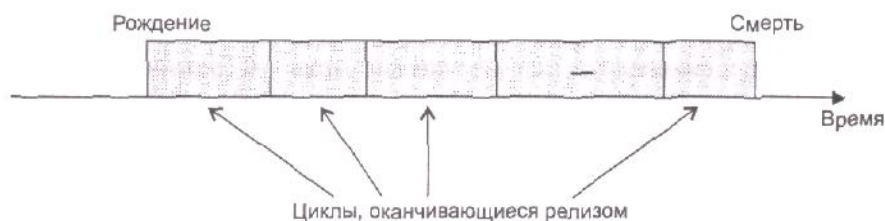
Эти концепции — управляемая вариантами использования, архитектурно-ориентированная и итеративная и инкрементная разработка — *одинаково важны*. Удаление одной из этих трех ключевых идей сильно *уменьшит ценность* Унифицированного процесса.

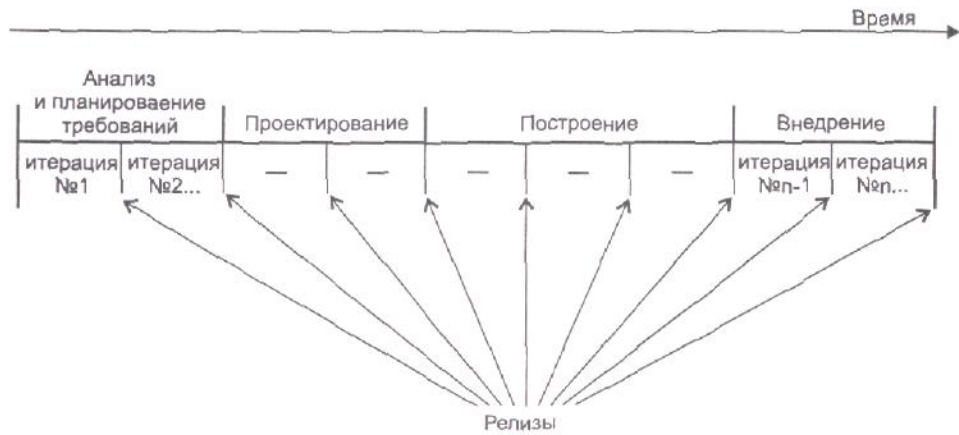
Архитектура предоставляет структуру, направляющую работу в итерациях, в каждой из которых варианты использования определяют цели и направляют работу.

Жизненный цикл Унифицированного процесса

Унифицированный процесс циклически *повторяется*. Последовательность повторений Унифицированного процесса представляет собой *жизненный цикл системы*.

Каждый цикл завершается поставкой **выпуска** продукта заказчику.





Каждый цикл состоит из четырех фаз:

1. анализа и планирования требований
2. проектирования
3. построения
4. внедрения

Каждая **фаза** подразделяется на *итерации* (см. рис.).

Разделение цикла разработки на фазы

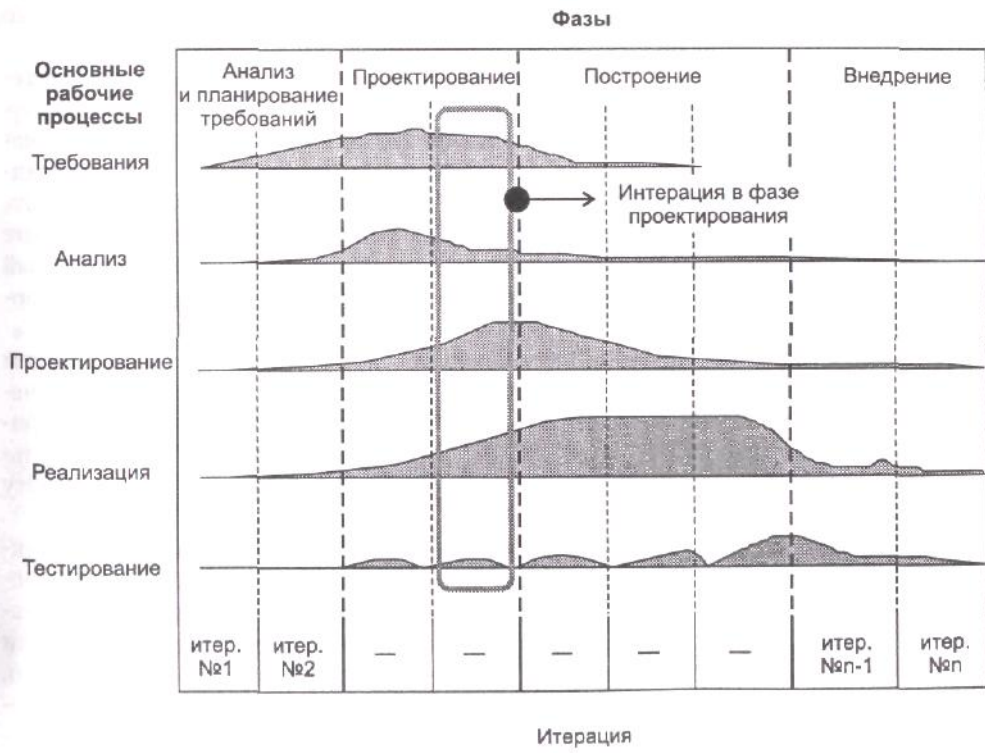
Каждый *цикл* осуществляется в течение некоторого времени.

Цикл *делится* на четыре *фазы*.

Каждая фаза заканчивается **вехой**. *Веха* определяется по наличию определенного набора артефактов.

Вехи служат нескольким целям. Некоторые из них следующие:

- дать руководителям возможность принять некоторые критические решения перед тем, как работа перейдет на следующую фазу;
- дать руководству и самим разработчикам возможность отслеживать прогресс в работе при проходах этих четырех ключевых точек;
- отслеживая время и усилия, затраченные на каждую фазу, создается массив данных.



Интегрированный процесс

Унифицированный процесс основан на компонентах. Он использует новый стандарт визуального моделирования — Унифицированный язык моделирования (UML), и базируется на трех ключевых идеях — вариантах использования, архитектуре и итеративной и инкрементной разработке. Необходим многоплановый процесс, поддерживающий:

- циклы
- фазы
- рабочие процессы
- снижение рисков
- контроль качества
- управление проектом и конфигурацией.

Унифицированный процесс создает *каркас*, позволяющий объединить все эти различные аспекты.

Четыре «П» — персонал, проект, продукт и процесс — в разработке программного обеспечения

Итогом проекта по разработке программного обеспечения является **продукт**, в создании которого принимает участие множество различных людей.

Процесс разработки программного обеспечения *направляет* усилия людей, вовлеченных в проект. Легче всего представить его в виде шаблона, предписывающего шаги, необходимые для выполнения проекта.

Обычно процесс *автоматизируется* при помощи утилиты или набора утилит (см. рис. ниже).

Понятия *персонал, проект, продукт, процесс* и *утилиты*, можно определить следующим образом:

- *Персонал*: архитекторы, разработчики, тестеры и их руководство, а также пользователи, заказчики и другие заинтересованные лица — являются исходной движущей силой программного проекта. Персонал — это реальные люди, в отличие от абстрактной конструкции *сотрудники*.
- *Проект*: организационная сущность, при помощи которой происходит управление разработкой программного обеспечения. Результат проекта — выпущенный *продукт*.
- *Продукт*: артефакты, создаваемые в течение жизни проекта, такие как **модели**, тексты программ, исполняемые файлы и документация.
- *Процесс*: т.е. процесс создания программного обеспечения — это *определение* полного набора *видов деятельности*, необходимых для преобразования требований пользователя в продукт. Процесс служит шаблоном для создания проекта.
- *Утилиты*: программы, используемые для автоматизации определенных в процессе видов деятельности.



Размещение «Ресурсов» внутри «Сотрудников»

В организации, производящей программное обеспечение, персонал заполняет множество различных позиций. Подготовка людей к занятию этих позиций включает соответствующее образование и специальную подготовку, сопровождаемую самостоятельной работой под присмотром преподавателей.

Здесь выбрано слово **сотрудник** для описания позиции, которая может быть поручена персоналу и которую может занимать работник. *Сотрудник* — это роль, которую может играть человек в разработке программного обеспечения, например: спецификатор вариантов использования, архитектор, инженер по компонентам или тестер интеграции.

Каждый сотрудник (то есть экземпляр сотрудника) соотносится с определенным набором видов деятельности, например, необходимых для разработки подсистемы. Для эффективной работы сотруднику необходима информация о том, как выполнять эти виды деятельности. У них должны быть адекватные утилиты.

Каждый сотрудник разработки программ имеет набор обязанностей и осуществляет набор видов деятельности.

При выделении ресурсов менеджер проекта должен минимизировать перебрасывание артефактов от одного ресурса к другому, чтобы процесс протекал настолько гладко, насколько это возможно.

Проект порождает продукт

Проект по разработке программного обеспечения заканчивается новым выпуском продукта. В первом проекте жизненного цикла (или первом цикле разработки, иногда называемом “целинным проектом”) разрабатывается и выпускается исходная система или продукт. В последовательных циклах проектирования система проходит череду множества выпусков.

В ходе жизненного цикла команда, работающая над проектом, сталкивается с изменениями, итерациями и организационными шаблонами, которыми будет определяться развитие проекта:

- **Последовательность изменений.** Результатом проекта по разработке программного обеспечения является продукт, но путь, которым идет разработка, есть серия изменений. Этот факт жизни проекта проявляется в умах разработчиков по мере продвижения по фазам и итерациям. Каждый цикл, каждая фаза и каждая итерация изменяют систему.

Первый цикл разработки — особый случай, превращающий систему из ничего во что-то. Каждый цикл завершается *выпуском*. Изменения продолжают появляться и на уровне, более высоком, чем последовательность циклов, в течение *поколений*.

- *Серии итераций*. Внутри каждой фазы или цикла деятельность сотрудников протекает в виде серий итераций. В ходе каждой итерации происходит реализация набора взаимосвязанных вариантов использования или уменьшение определенных рисков. В ходе итерации разработчики проделывают серию рабочих процессов — требования, проектирование, реализация и тестирование. В любой итерации выполняются все эти рабочие процессы, мы рассматриваем итерацию как *мини-проект*.
- *Организационный шаблон*. Проект включает в себя команду разработчиков, которые должны получить результат, не выходя при этом за пределы бизнес-ограничений, таких, как время, деньги и качество. Люди принимают на себя функции различных сотрудников. Идея “процесса” — создать образец, внутри которого люди, исполняя функции сотрудников, выполняют проект. Этот шаблон или образец определяет типы сотрудников, необходимых для работы над проектом, и артефакты, с которыми они должны работать. Процесс также содержит множество подсказок, эвристик и примеров документирования, которые помогают назначенному персоналу делать свою работу.

Продукт — это больше, чем код

В контексте Унифицированного процесса разрабатываемый продукт — это программная система. Понятие *продукта* не ограничивается поставляемым кодом, а относится ко *всей системе целиком*.

Что такое программная система?

Системой являются все артефакты, существующие в понятном компьютеру или человеку виде и представляемые компьютерам, сотрудникам или заинтересованным лицам.

Система содержит набор моделей

Наиболее интересный тип артефактов, с которыми имеет дело Унифицированный процесс, — это модели. Каждый сотрудник нуждается в своем уникальном видении системы (см. рис. ниже). При разработке Унифицированного процесса авторами были идентифицированы все сотрудники и все варианты обозрения системы, которые могут им понадобиться. Отобранные варианты обозрения системы для всех сотрудников были структурированы в нечто большее — в модели, чтобы сотрудник мог получить из набора моделей любой конкретный вариант обозрения.

Построение системы, таким образом, это построение моделей и использование разных моделей для описания *различных сторон системы*. Выбор модели для системы — это одно из наиболее важных решений, принимаемых командой разработчиков.

Что такое модель?

Модель — это абстракция, описывающая моделируемую систему с определенной точки зрения и на определенном уровне абстрагирования.

Модели — это абстракции системы, которые создаются архитекторами и проектировщиками.

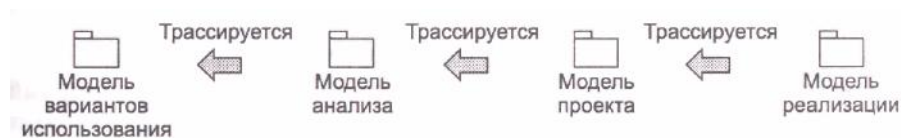
Модель изнутри

Модель всегда отождествляется с моделируемой системой. Этот элемент системы служит затем контейнером для других элементов. Подсистема верхнего уровня соответствует создаваемой системе.

Связи между моделями

Система содержит все связи между элементами модели и все ограничения, а также разграничения между элементами различных моделей. Итак, система — это не просто набор моделей, но и набор связей между ними.

Так, каждый вариант использования в модели вариантов использования имеет связь с соответствующим элементом модели анализа (и т. д.). Эта связь в UML называется связью трассировки или просто **трассировкой**.



Процесс направляет проекты

Понятие *процесс* трактуется по-разному. Оно употребляется в различном смысле для различных контекстов, таких как бизнес-процесс, процесс разработки, программный процесс. В контексте Унифицированного процесса мы считаем ключевым “бизнес”–процессом разработку программного обеспечения, то есть организацию разработки и сопровождения программ (или разработку бизнеса по производству программного обеспечения). В этом бизнесе существуют и другие процессы, например процесс сопровождения, характеризуемый взаимодействием с пользователями продукта, и процесс продаж, который начинается с получения заказа и оканчивается поставкой продукта.