

Стандарты жизненного цикла разработки ИС -2

Лектор – Скороход С. В.

ГОСТ Р ИСО/МЭК 12207:2010

Русскоязычный вариант стандарта **ISO/IEC 12207:2008**

- Стандарт на процессы в организации, в том числе и организацию жизненного цикла.
- Стандарт не содержит описания фаз, стадий, этапов, форм документов. Содержит только описание процессов.
- Все процессы ЖЦ программного обеспечения ИС делятся на три группы:
 - Основные. Это процессы, непосредственно относящиеся к жизненному циклу информационной системы. Можно считать, что это производственные процессы организации.
 - Вспомогательные. Это процессы, предназначенные для поддержки основных процессов. Сами по себе эти процессы организации не нужны - только в связи с основными процессами, которые они обслуживают.
 - Организационные. Это общекорпоративные процессы, такие как "Обучение" или "Управление". Эти процессы существуют в организации независимо от того, как организовано производство и как устроены вспомогательные процессы.

ГОСТ Р ИСО/МЭК 12207: Структура процессов в организации

5. Основные процессы жизненного цикла

5.1 Заказ

5.2 Поставка

5.3
Разработка

5.4
Эксплуатация

5.5
Сопровождение

6. Вспомогательные процессы жизненного цикла

6.1 Документирование

6.2 Управление
конфигурацией

6.3 Обеспечение качества

6.4 Верификация

6.5 Аттестация

6.6 Совместный анализ

6.7 Аудит

6.8 Решение проблем

7. Организационные процессы жизненного цикла

7.1 Управление

7.2 Создание инфраструктуры

7.3 Усовершенствование

7.4 Обучение

ГОСТ Р ИСО/МЭК 12207: Структура процессов в организации

- Если на протяжении жизненного цикла программной системы возникает задача, которая имеет общекорпоративный характер, например задача обучения, используется соответствующий процесс из стандарта.
- Процессы состоят из работ, работы - из задач. Последовательность работ и задач, приведенная в стандарте, не является жесткой и может быть изменена. Необходимо только выдерживать логические связи между работами и задачами.
- Цель стандарта - определить полную совокупность процессов, которые могут выполняться в ходе проекта по созданию программной системы. Но поскольку проекты могут сильно различаться, допускается для каждого проекта локально видоизменять использующиеся в нем процессы, исключая или добавляя отдельные работы и задачи. Такая деятельность называется в стандарте адаптацией. Стандарт содержит описание процесса собственной адаптации.

ГОСТ Р ИСО/МЭК 12207: Пример описания процесса

5.1. Процесс заказа

Процесс заказа состоит из работ и задач, выполняемых заказчиком. Процесс начинается с определения потребностей заказчика в системе, программном продукте или программной услуге. Далее следуют подготовка и выпуск заявки на подряд, выбор поставщика и управление процессом заказа вплоть до завершения приемки системы, программного продукта или программной услуги. Конкретная организация, имеющая соответствующую потребность, может быть названа собственником. Собственник может заключить договор на выполнение части или всех работ по заказу с посредником, который будет поочередно проводить данные работы в соответствии с процессом заказа. В данном подразделе под заказчиком понимается собственник или посредник. Заказчик управляет процессом заказа на проектном уровне в соответствии с процессом управления (7.1), который конкретизируется в данном процессе; определяет инфраструктуру для данного процесса в соответствии с процессом создания инфраструктуры (7.2); адаптирует данный процесс к условиям проекта в соответствии с процессом адаптации (Приложение А) и управляет процессом заказа на организационном уровне в соответствии с процессами усовершенствования (7.3) и обучения (7.4).

ГОСТ Р ИСО/МЭК 12207: Пример описания процесса

Список работ. Данный процесс состоит из следующих работ:

1. подготовка;
2. подготовка заявки на подряд;
3. подготовка и корректировка договора;
4. надзор за поставщиком;
5. приемка и закрытие договора.

Скороход Сергей Васильевич

ГОСТ Р ИСО/МЭК 12207: Пример описания процесса

5.1.1. Подготовка

Данная работа состоит из следующих задач:

5.1.1.1. Заказчик начинает процесс заказа, описывая концепцию или потребность в заказе, разработке или модернизации системы, программного продукта или программной услуги.

5.1.1.2. Заказчик должен определить и проанализировать требования к системе. Требования к системе должны охватывать функциональные, коммерческие, организационные, потребительские аспекты системы, а также требования к безопасности, защите и другие критические требования наряду с требованиями к проектированию, тестированию и соответствующим стандартам и процедурам.

5.1.1.3. Если заказчик поручает поставщику выполнение анализа требований к системе, то заказчик должен согласовать требования, сформулированные в результате анализа.

5.1.1.4. Заказчик может выполнить определение и анализ требований к программным средствам сам или поручить решение этой задачи поставщику.

ГОСТ Р ИСО/МЭК 12207: Пример процесса

5.1.1.5. При решении задач, определенных в 5.1.1.2 и 5.1.1.4, должен использоваться процесс разработки (подраздел 5.3).

5.1.1.6. Заказчик должен рассмотреть варианты реализации заказа начиная с анализа соответствующих критериев, включая рискованность и стоимость проекта и выгоды от каждого варианта.

Анализируются следующие варианты:

- а) покупка готового программного продукта, удовлетворяющего определенным требованиям;
- б) разработка программного продукта или получение программной услуги собственными силами;
- с) разработка программного продукта или получение программной услуги на договорной основе;
- д) комбинации по перечислениям а), б), с);
- е) модернизация существующего программного продукта или услуги.

ГОСТ Р ИСО/МЭК 12207: Пример процесса

5.1.1.7. При приобретении готового программного продукта заказчик должен получить гарантии того, что удовлетворены следующие условия:

- а) программный продукт соответствует установленным требованиям;
- б) имеется в наличии соответствующая документация;
- с) соблюдены права собственности, использования, лицензирования и гарантии;
- д) предусмотрена последующая поддержка программного продукта.

5.1.1.8. Заказчик должен подготовить, документально оформить и выполнить план заказа. План должен содержать:

- а) требования к системе;
- б) планируемую загрузку системы;
- с) тип реализуемого договора;
- д) обязанности организаций, участвующих в договоре;
- е) обеспечение подходов к реализации договора;
- ф) анализ возможных рискованных ситуаций, а также методы управления такими ситуациями.

5.1.1.9. Заказчик должен определить и документально оформить принятые правила и условия (критерии) реализации договора.

ГОСТ Р ИСО/МЭК 12207: Особенности

- Описательный характер определения процессов.
- Скорее рекомендательный, чем директивный порядок работ.
- Полное отсутствие форматов документов, возникающих в ходе выполнения работ и задач. Это, с одной стороны, оставляет свободу при внедрении стандарта, с другой - резко затрудняет процесс внедрения.
- Фактически управленцы должны будут самостоятельно разработать формы основных документов, согласовать и утвердить их у руководства.
- Внедрение стандарта требует значительно более высокой квалификации управленцев и более эффективной работы организации в целом.
- Внедрение стандарта остается пока уделом немногих продвинутых организаций и предприятий.

Методологии, регламентирующие ЖЦ ПО ИС

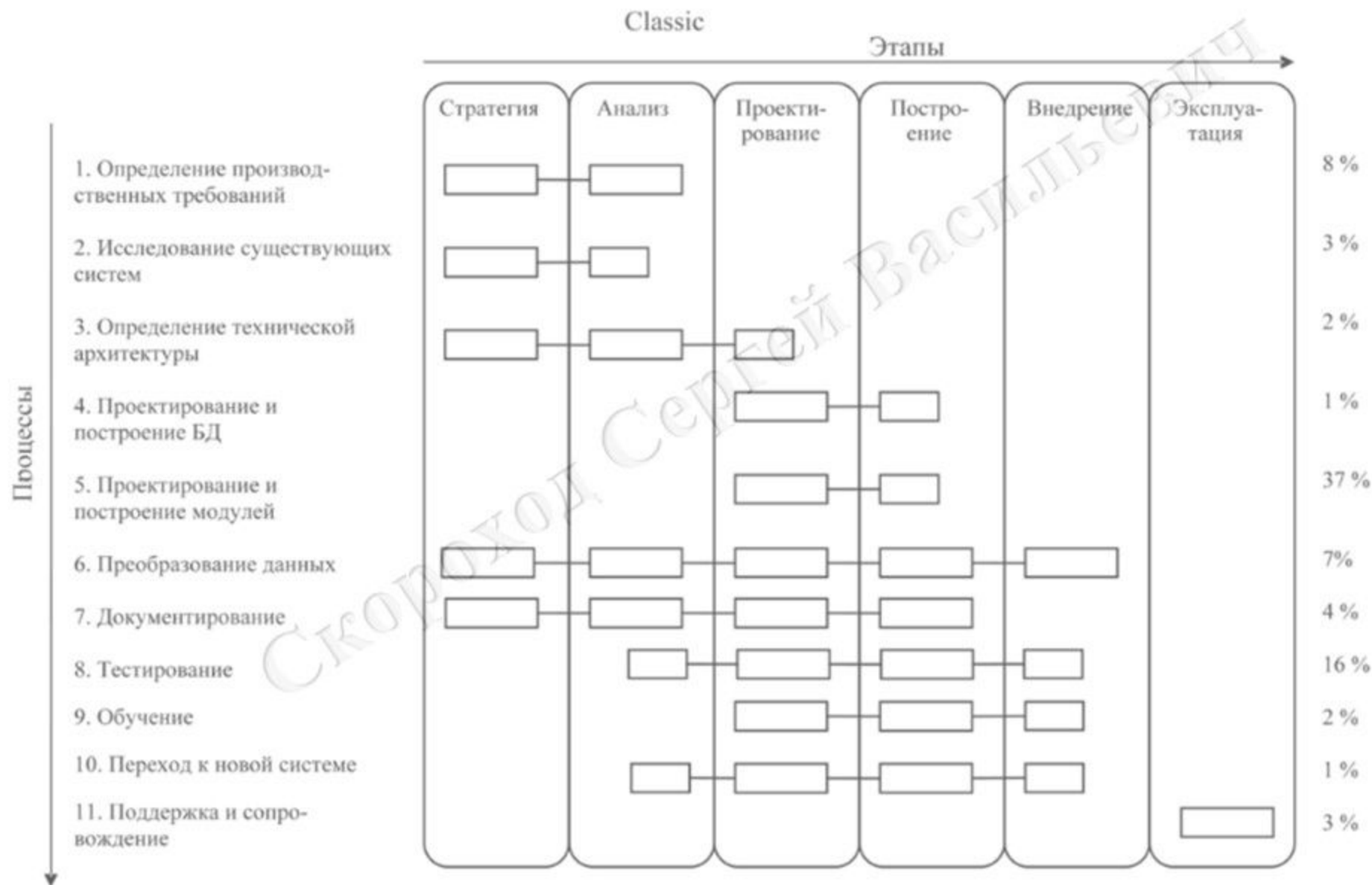
1. **Custom Development Method (CDM)** - методика Oracle.
2. **Rational Unified Process (RUP)** – методика IBM.
3. **Microsoft Solution Framework MSF** (MicroSoft).

Скороход Сергей Васильевич

Custom Development Method (CDM)

- Технологический материал, детализированный до уровня заготовок проектных документов, рассчитанных на прямое использование в проектах ИС с опорой на инструментарий фирмы Oracle.
- Совокупность точно определенных процессов заказной разработки с разными режимами управления.
- Методология, в основе которой лежит CASE-технология, обеспечивает точное определение бизнес-требований в самом начале процесса разработки и их сохранение на протяжении всего процесса разработки.
- Методология CDM радикально повышает возможность успешной реализации проекта.
- Применяется для классической (Classic) модели ЖЦ (предусмотрены все работы/задачи и этапы), а также для технологий "быстрой разработки" (Fast Track) или "облегченного подхода" (Light), рекомендуемых в случае малых проектов.

Custom Development Method (CDM)



Custom Development Method (CDM)

- Модель ЖЦ методологии Oracle CDM имеет два измерения.
- **Первое измерение** связано с тем, какой процесс должен быть выполнен для разработки проекта. Это измерение определяется процессами в рамках CDM.
- **Второе измерение** связано с тем, когда должны выполняться процессы в ЖЦ проекта. Это измерение определяется этапами ЖЦ.



Custom Development Method (CDM)

- Для каждой модели ЖЦ существует свой набор этапов и процессов.
- Модель ЖЦ **Classic** имеет шесть этапов и одиннадцать процессов. Классический подход в CDM применяется для крупных разработок продолжительностью более восьми месяцев и содержит шесть этапов: определение требований (стратегия), анализ, проектирование, реализация, внедрение и эксплуатация.
- Модель **Fast Track** предназначена для заказных разработок среднего объема (продолжительностью 6—12 месяцев) и содержит только три этапа (моделирование требований, проектирование и создание системы, внедрение) и все 11 процессов, как и в модели Classic.
- Модель **Lite** (облегченная разработка) применяется для небольших проектов продолжительностью до полугода и использует всего два этапа (прототипирование и построение; внедрение) и девять процессов (кроме 6-го и 3-го).

Rational Unified Process (RUP)

- Использует итеративную модель разработки. В конце каждой итерации (в идеале продолжающейся от 2 до 6 недель) проектная команда должна достичь запланированных на данную итерацию целей, создать или доработать проектные артефакты и получить промежуточную, но функциональную версию конечного продукта.
- Каждая фаза может быть разбита на этапы (итерации), в результате которых выпускается версия для внутреннего или внешнего использования.
- Выделяют 4 основные фазы: начало, уточнение (развитие), построение (конструирование), передача.
- Прохождение через все четыре основные фазы называется циклом разработки, каждый цикл завершается генерацией версии системы.
- Если после этого работа над проектом не прекращается, то полученный продукт продолжает развиваться и снова минует те же фазы.

Rational Unified Process (RUP)

Основные принципы:

- 1 — ранняя идентификация и непрерывное (до окончания проекта) устранение основных рисков;
- 2 — концентрация на выполнении требований заказчиков к исполняемой программе;
- 3 — ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки;
- 4 — компонентная архитектура, реализуемая и тестируемая на ранних стадиях проекта;
- 5 — постоянное обеспечение качества на всех этапах разработки проекта (продукта);
- 6 — работа над проектом в сплочённой команде, ключевая роль в которой принадлежит архитекторам.

Rational Unified Process (RUP)



- Жизненный цикл программного обеспечения состоит из четырех последовательных фаз (этапов), в конце каждой из которых находится большая веха.
- Таким образом, каждый этап представляет собой промежуток времени между двумя большими вехами.
- В конце каждого этапа выполняется проверка того, были ли достигнуты цели, которые были поставлены для данного этапа.
- Если будет получен положительный ответ, начинается выполнение следующего этапа проекта.

Rational Unified Process (RUP)

Фаза начало

- Формируются видение и границы проекта.
- Создается экономическое обоснование (business case).
- Определяются основные требования, ограничения и ключевая функциональность продукта.
- Создается базовая версия модели прецедентов.
- Оцениваются риски.
- Команда решает, стоит ли вообще заниматься этим проектом, исходя из его предполагаемой стоимости, необходимых ресурсов и цели, которую нужно достичь.

Rational Unified Process (RUP)

Фаза уточнение

- Производится анализ предметной области и построение исполняемой архитектуры.
- Документирование требований (включая детальное описание для большинства прецедентов).
- Спроектированная, реализованная и оттестированная исполняемая архитектура.
- Обновленное экономическое обоснование и более точные оценки сроков и стоимости.
- Сниженные основные риски.

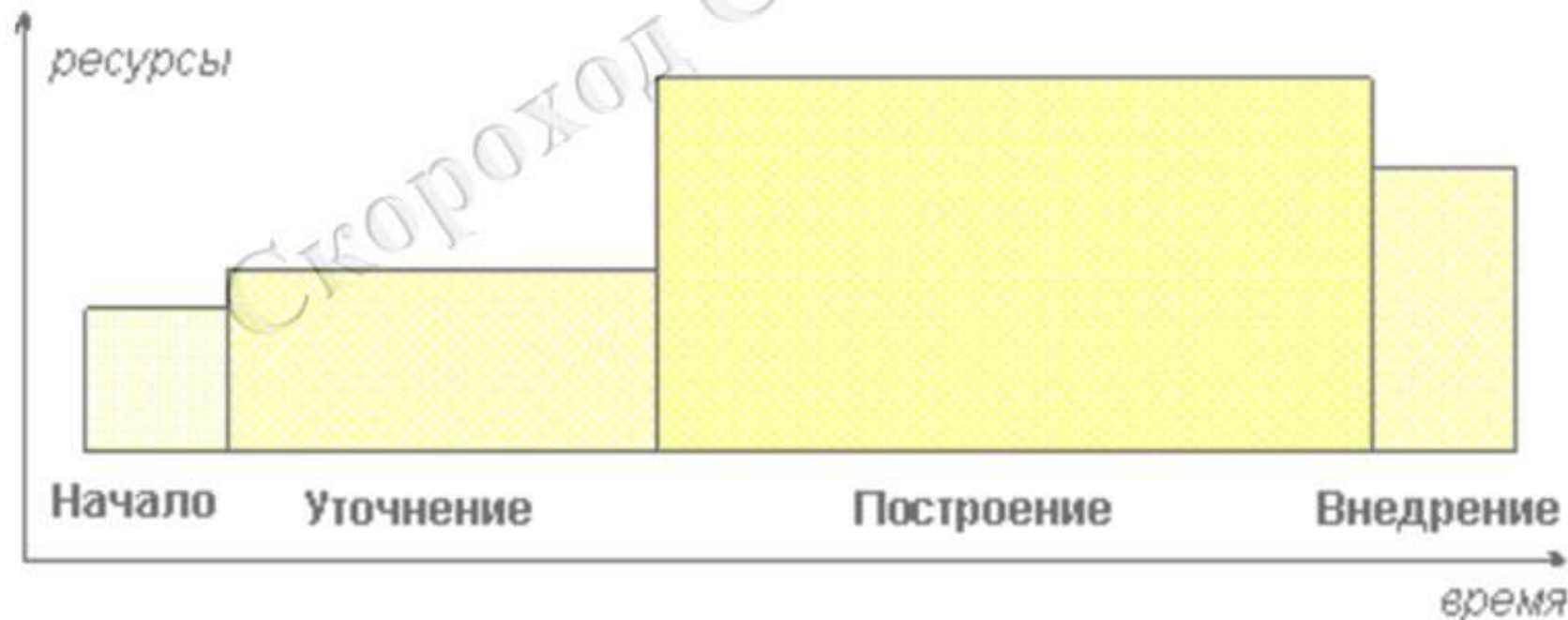
Фаза построение

- Происходит реализация большей части функциональности продукта.

Rational Unified Process (RUP)

Фаза внедрение

- Создается финальная версия продукта и передается от разработчика к заказчику.
- Это включает в себя программу бета-тестирования, обучение пользователей, а также определение качества продукта.
- В случае, если качество не соответствует ожиданиям пользователей или критериям, установленным в фазе Начало, фаза Внедрение повторяется снова.



Rational Unified Process (RUP)

Основные процессы RUP (6 процессов)

1. **Моделирование предметной области** - понять бизнес-контекст, в котором должна будет работать система, понять возможные проблемы, оценить возможные их решения и их последствия для бизнеса и организации заказчика.
2. **Определение требований** - понять, что должна делать система, определить границы системы и основу для планирования проекта и оценок ресурсозатрат в нем.
3. **Анализ и проектирование** - выработка архитектуры системы на основе ключевых требований, создание проектной модели, представленной в виде диаграмм UML, описывающих продукт с различных точек зрения.
4. **Реализация** - разработка исходного кода, компонент системы, тестирование и интегрирование компонент.
5. **Тестирование** - Общая оценка дефектов продукта, его качество в целом; оценка степени соответствия исходным требованиям.
6. **Развертывание** - развернуть систему в ее рабочем окружении и оценить ее работоспособность.

Rational Unified Process (RUP)

Вспомогательные процессы RUP (3 процесса)

- 1. Управление конфигурациями и изменениями** - Определение элементов, подлежащих хранению и правил построения из них согласованных конфигураций, поддержание целостности текущего состояния системы, проверка согласованности вносимых изменений.
- 2. Управление проектом** - Включает планирование, управление персоналом, обеспечения связей с другими заинтересованными лицами, управление рисками, отслеживание текущего состояния проекта.
- 3. Управление средой проекта** - Настройка процесса под конкретный проект, выбор и смена технологий и инструментов, используемых в проекте.

Rational Unified Process (RUP)

Рабочие процессы

Стадии

Основные процессы

Бизнес-моделирование

Управление требованиями

Анализ и проектирование

Реализация

Тестирование

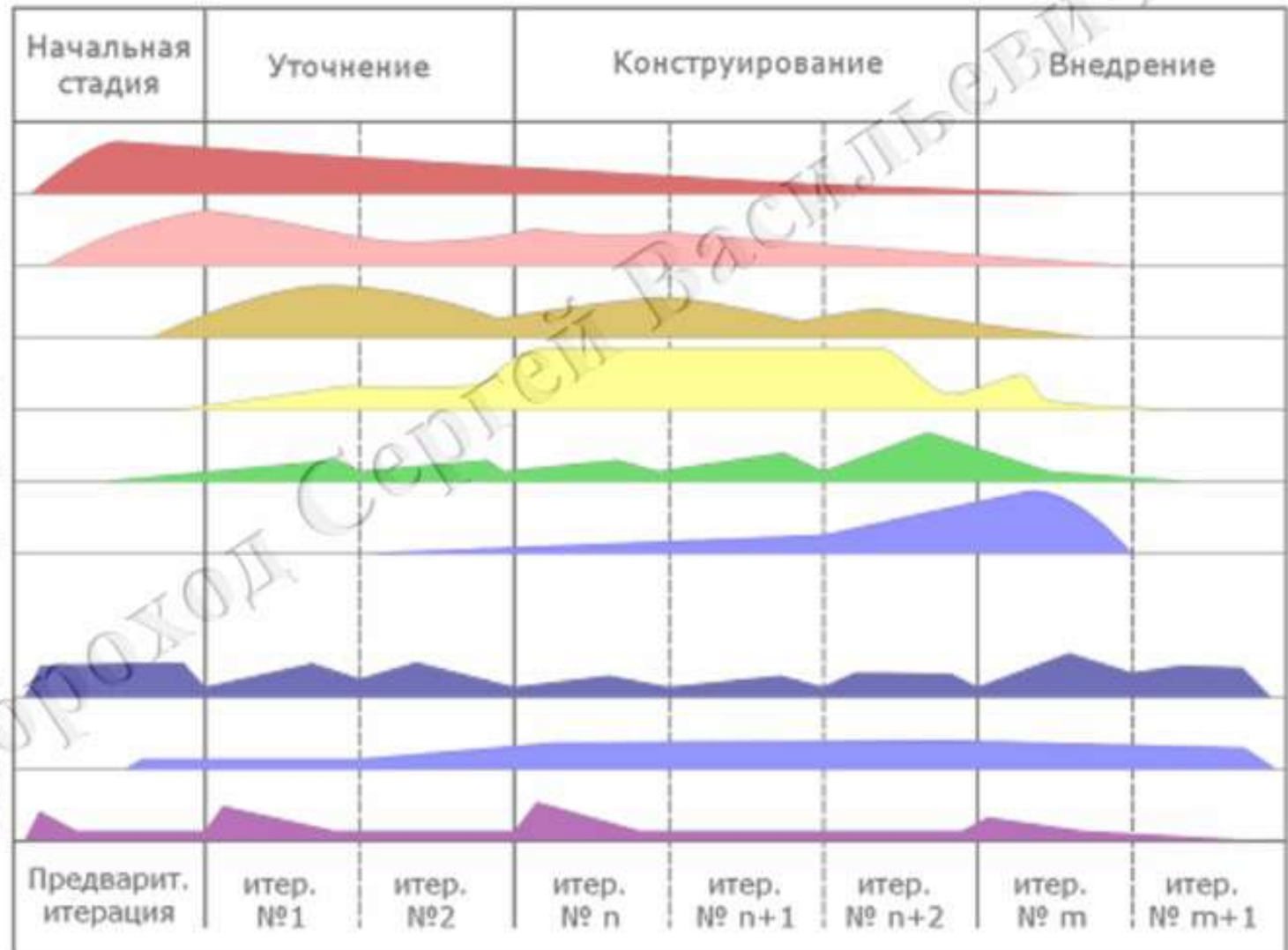
Развертывание

Поддерживающие процессы

Управление проектом

Управление конфигурацией
и изменениями

Создание инфраструктуры
(среда разработки)



Итерации

Rational Unified Process (RUP)

Достоинства RUP

- Методология RUP позволяет справляться с изменениями в требованиях, независимо от того, исходят они от клиента или возникают в ходе работы над проектом;
- RUP подчёркивает необходимость точной документации.
- Интеграция требований происходит на протяжении всего процесса разработки, и в частности в фазе построения.

Недостатки RUP

- RUP опирается на способность экспертов и профессионалов назначить действия определённым работникам, которые затем обязаны выдать запланированные результаты в виде артефактов.
- Интеграция в процесс разработки может негативно сказаться на другой более фундаментальной деятельности на этапах тестирования.
- Это довольно сложный метод, который трудно внедрить в свой проект, особенно если у вас небольшая команда или компания.