

ВВЕДЕНИЕ
В КУРС «МЕНЕДЖМЕНТ В
ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЯХ»

Введение

Введение в курс «Менеджмент в информационных технологиях»

Курс «Менеджмент в ИТ» рассматривает вопросы управления .

Последний экономический кризис заставил многие организации урезать проектные бюджеты. Но так как проекты все еще нужно кому-то завершать, появился новый тренд совмещать в одном человеке роли менеджера проекта (PM) и бизнес аналитика (BA). Назвали эту роль проектный профессионал «project professional» (PP).

Новая проектная роль $PP = PM + BA$

От руководителей проектов начали все больше требовать сбор требований и их анализ. От бизнес аналитиков управлять большим количеством проектов. И это все помимо основных обязанностей.

Введение

Содержание курса

Введение

Проектный профессионал – новая проектная роль. Содержание, цели и задачи курса, обзор литературы.

Тема 1. Введение в программную инженерию

Менеджмент в ИТ, его составные части, основные термины и определения в программной инженерии. Жизненный цикл ПО и его модели. Основные и дополнительные области знаний.

Тема 2. Модели процесса разработки ПО

Классификация моделей. CMMI (Capability Maturity Model Integration), стандарты IEEE/ISO, модель RUP (Rational Unified Process), «гибкие» модели Agile (Scrum).

Тема 3. Принципы управление проектами

Основные термины и определения. Тройственная ограниченность. Подходы и методы: PERT (Program (Project) Evaluation and Review Technique), диаграмма Ганта.

Введение

Тема 4. Инициация проекта

Требования и их анализ, концепция проекта, цели и результаты проекта, ключевые участники и заинтересованные стороны, ресурсы, сроки, риски, критерии приемки.

Тема 5. Планирование проекта

Уточнение содержания и состава работ. Планирование организационной структуры. Планирование управления конфигурациями. Планирование управления качеством. Управление рисками, оценка и контроль проекта.

Тема 6. Реализация проекта

Управление требованиями. Принципы количественного управления. Завершение проекта.

Список рекомендуемой литературы

1. «PMBOK. Руководство к Своду знаний по управлению проектами», 3-е изд., PMI, 2004.
2. «Руководство к своду знаний по программной инженерии». The Guide to the Software Engineering Body of Knowledge, SWEBOK, IEEE Computer Society Professional Practices Committee, 2004.
3. Скотт Беркун, «Искусство управления IT-проектами», пер. с англ., СПб., Питер-Пресс, 2007. – 400 с. ISBN 978-5-91180-005-5
4. Сергей Архипенков «Лекции по управлению программными проектами», 2009.
5. Брукс Фредерик, "Мифический человеко-месяц, или Как создаются программные комплексы", Пер. с англ., СПб., Символ-Плюс, 1999.
6. А. Якобсон, Г. Буч, Дж. Рамбо Унифицированный процесс разработки программного обеспечения. – СПб.: ПИТЕР, 2002. – 496с. ISBN 5-318-00358-3

Цели и задачи курса

МЕНЕДЖМЕНТ В ПРОГРАММНОЙ ИНЖЕНЕРИИ

Задачи:

- Ознакомиться с основными принципами управления проектами;
- Ознакомиться с основными процессами разработки ПО;
- Ознакомиться с основными ролями разработчиков ПО.

Тема 1
ВВЕДЕНИЕ В ПРОГРАММНУЮ
ИНЖЕНЕРИЮ

Тема1. Введение в программную инженерию

Менеджмент (англ. *management*) – управление, руководство, администрирование, дирекция, регулирование

Составные части менеджмента в ИТ

Крупная ИТ-компания одновременно ставит и решает комплекс взаимосвязанных задач, для чего создается несколько подсистем в системе менеджмента:

- Менеджмент программной инженерии:
 - Управление жизненным циклом программного продукта
 - Управление персоналом
 - Управление качеством
- Маркетинг
- Финансовый менеджмент
- Стратегический менеджмент
- Инвестиционный менеджмент
- Инновационный менеджмент
- Риск менеджмент
- Информационный менеджмент
- Экологический менеджмент

Тема 1. Введение в программную инженерию

ОСНОВНЫЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Программная инженерия есть применение определенного систематического измеримого подхода при разработке, эксплуатации и поддержке программного обеспечения [1].

Термин software engineering (программная инженерия) впервые появился в названии конференции НАТО, состоявшейся в Германии в 1968 году и посвященной так называемому кризису программного обеспечения. С 1990-го по 1995 год велась работа над международным стандартом, который должен был дать единое представление о процессах разработки программного обеспечения. В результате был выпущен стандарт ISO/IEC 12207 [2].

В 2004 году в отрасли был создан основополагающий труд «Руководство к своду знаний по программной инженерии» (SWEBOK) [3], в котором были собраны основные теоретические и практические знания, накопленные в этой отрасли.

Тема 1. Введение в программную инженерию

Программирование — процесс отображения определенного множества целей на множество машинных команд и данных, интерпретация которых на компьютере или вычислительном комплексе обеспечивает достижение поставленных целей.

Программный продукт — совокупность программ и сопроводительной документации по их установке, настройке, использованию и доработке.

Процесс разработки ПО — совокупность процессов, обеспечивающих создание и развитие программного обеспечения.

Модель процесса разработки ПО — формализованное представление процесса разработки ПО.

Тема 1. Введение в программную инженерию

Жизненный цикл проекта (Project Life Cycle Management – PLCM)

Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания программного продукта и заканчивается в момент его полного изъятия из эксплуатации^[1]. Этот цикл — процесс построения и развития ПО.

В общем случае, жизненный цикл определяется **моделью** и описывается в форме **методологии** (метода). Модель или парадигма жизненного цикла определяет концептуальный взгляд на организацию жизненного цикла и, часто, основные фазы жизненного цикла и принципы перехода между ними.

Модель жизненного цикла ПО — структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач на протяжении жизненного цикла. Модель жизненного цикла зависит от специфики, масштаба и сложности проекта и специфики условий, в которых система создается и функционирует.

Тема 1. Введение в программную инженерию



Тема 1. Введение в программную инженерию

Методология (метод) задает комплекс работ, их детальное содержание и ролевую ответственность специалистов на всех этапах выбранной модели жизненного цикла, обычно определяет и саму модель, а также рекомендует практики (best practices), позволяющие максимально эффективно воспользоваться соответствующей методологией и ее моделью.

Ниже приведены определения модели жизненного цикла программной системы, даваемые, например, в различных вариантах стандартов ГОСТ:

- Модель жизненного цикла - структура, состоящая из процессов, работ и задач, включающих в себя разработку, эксплуатацию и сопровождение программного продукта, охватывающая жизнь системы от установления требований к ней до прекращения ее использования [ГОСТ 12207, 1999].
- Жизненный цикл автоматизированной системы (АС) - совокупность взаимосвязанных процессов создания и последовательного изменения состояния АС, от формирования исходных требований к ней до окончания эксплуатации и утилизации комплекса средств автоматизации АС [ГОСТ 34, 1990].

Тема 1. Введение в программную инженерию

Процессы жизненного цикла программного обеспечения (Software Life Cycle Processes)

Определения процессов жизненного цикла обычно являются более детальными, чем модели. Однако, определения процессов не описывают порядка их выполнения во времени (за это как раз и отвечают модели, прим. автора). Это означает, что, в принципе, процессы жизненного цикла программного обеспечения могут быть “выстроены” (во времени) соответственно любой модели жизненного цикла. Основным источником знаний по процессам является стандарт IEEE/ISO/ГОСТ 12207 “Information Technology – Software Lifecycle Processes”

Тема 1. Введение в программную инженерию

Согласно SWEBOOK 2004, программная инженерия включает в себя 10 основных и 7 дополнительных областей знаний, на которых базируются процессы разработки. К основным областям знаний относятся следующие:

- **Software requirements** — программные требования: выявление, анализ, спецификация и проверка требований к программному обеспечению;
- **Software design** — дизайн (архитектура): процесс определения архитектуры, компонентов, интерфейсов и других характеристик системы или компонента;
- **Software construction** — конструирование ПО: поэтапное создание работающего программного обеспечения;
- **Software testing** — тестирование: динамический контроль поведения программы на конечном множестве тестов, надлежащим образом выбранных из бесконечной области;
- **Software maintenance** — эксплуатация (поддержка) ПО: совокупность мероприятий, необходимых для обеспечения экономически эффективной поддержки программного обеспечения;
- **Software engineering process** — процессы программной инженерии: определение, реализация, оценка, измерение, управление.

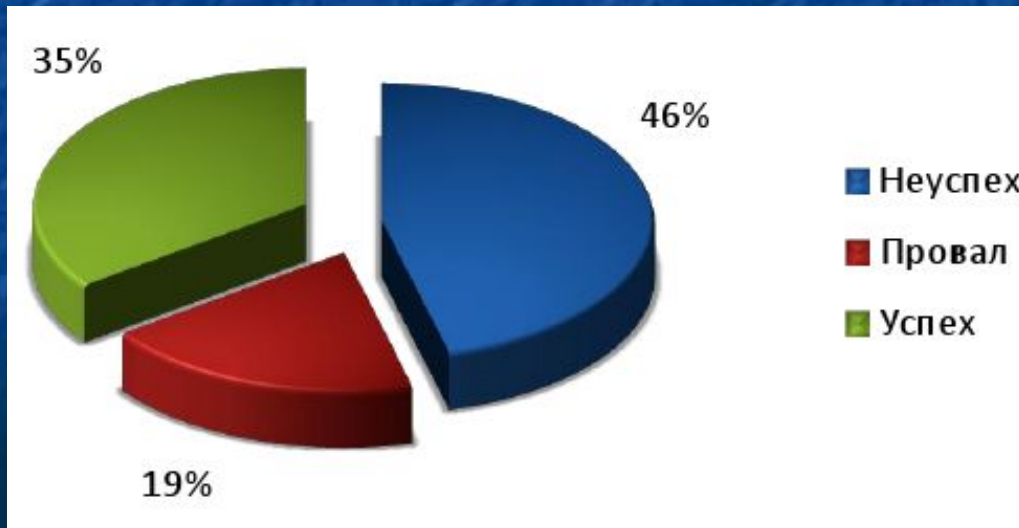
Тема 1. Введение в программную инженерию

- **Software configuration management** — конфигурационное управление: определение конфигурации системы на различные моменты времени для систематического контроля изменений конфигурации, а также сохранение целостности и прослеживаемости конфигурации на протяжении всего жизненного цикла системы;
- **Software engineering management** — управление в программной инженерии: применение мер управления, планирования, координации, измерения, мониторинга, контроля и отчетности, для того, чтобы разработка и сопровождение программного обеспечения являлась систематической и дисциплинированной;
- **Software engineering tools and methods** — инструменты и методы: компьютерные средства, которые предназначены для оказания помощи процессам жизненного цикла программы, а также методы, которые применяют к структуре деятельности разработки ПО с целью сделать разработку более систематической, чтобы иметь более шансов на успех;
- **Software quality** — управление качеством ПО: проверка удовлетворения требованиям к ПО набора собственных характеристик программы.

Тема 1. Введение в программную инженерию

Дополнительные области знаний включают в себя:

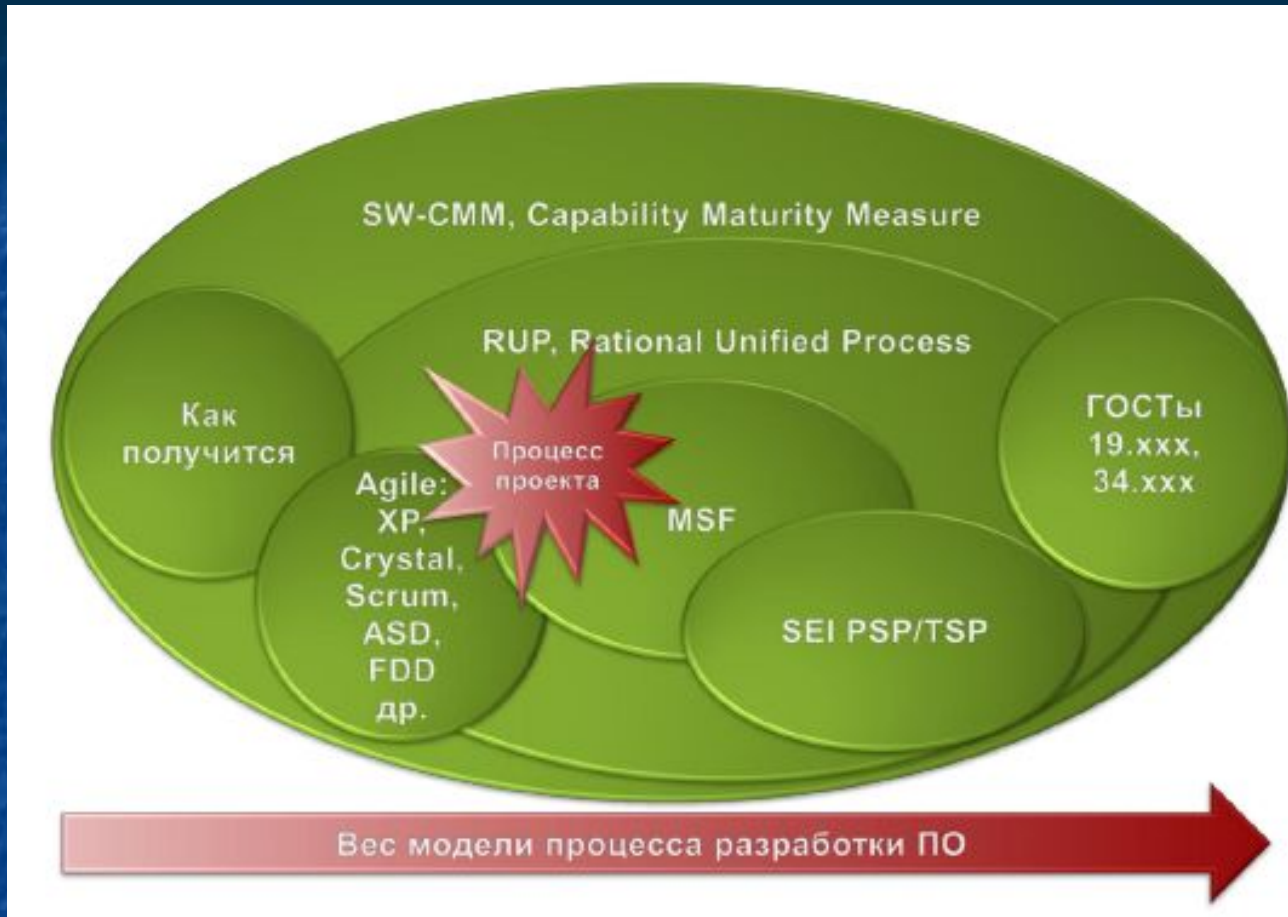
- Computer engineering — разработка компьютеров.
- Computer science — информатика.
- Management — общий менеджмент.
- Mathematics — математика.
- Project management — управление проектами.
- Quality management — управление качеством.
- Systems engineering — системное проектирование.



*Результаты анализа
успешность программных
проектов за 2006 год (по
данным Standish Group)*

ТЕМА 2
МОДЕЛИ ПРОЦЕССА
РАЗРАБОТКИ ПО

Тема 2. Модели процесса разработки ПО



+ ITIL
+ PMBOOK

Модели (методологии) процессов разработки ПО принято классифицировать по «**весу**» — количеству формализованных процессов (большинство процессов или только основные) и детальности их регламентации. Чем больше процессов документировано, чем более детально они описаны, тем больше «вес» модели.

Тема 1. Модели процесса разработки ПО

- **СТБ ИСО 9001-2009**
- **СММІ**
- **Отраслевые стандарты**
- **ITIL**
- **PMBOOK**

(Guide to the Project Management Body of Knowledge) - это проект «Project Management Institute», вобравший в себя накопленные знания в области управления проектами. Последняя версия документа вышла в 2000 году и тогда же получила статус стандарта американского института стандартизации ANSI (хотя стандарты ANSI и IEEE формально считаются американскими, большинство из них носит де-факто международный характер). Важной особенностью PMBoK является то, что он рассматривает управление проектами в общем смысле, без привязки к конкретным предметным областям, таким как ИТ.

Тема 2. Модели процесса разработки ПО

- ГОСТ 19 «Единая система программной документации», ГОСТ 34 «Стандарты на разработку и сопровождение автоматизированных систем»

Ориентированы на последовательный подход к разработке ПО.

Разработка в соответствии с этими стандартами проводится по этапам, каждый из которых предполагает выполнение строго определенных работ, и завершается выпуском достаточно большого числа весьма формализованных и обширных документов. Строгое следование этим ГОСТам приводит к водопадному подходу и требует очень высокой степени формализованности разработки. На основе этих стандартов разрабатываются программные системы по госзаказам

- СТБ ИСО 9001-2009

Предусматривают, с одной стороны, построение организационной системы "сверху вниз": от целей предприятия и его политики - к организационной структуре и формированию бизнес процессов, и с другой - итеративное развитие организационной системы через

Тема 2. Модели процесса разработки ПО

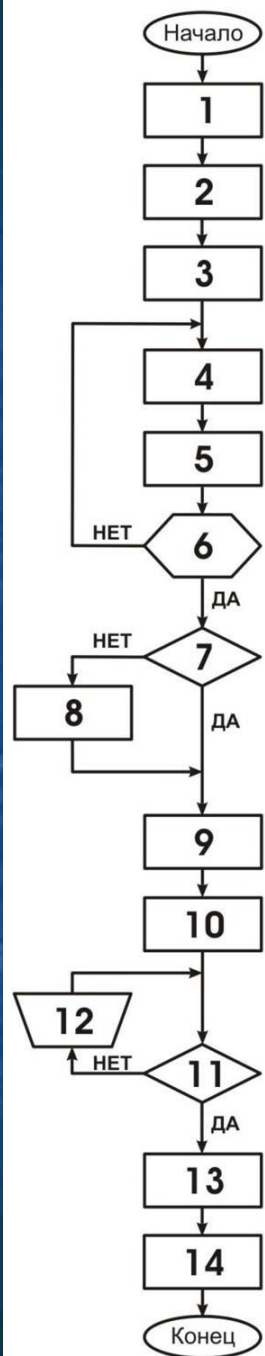
Упрощенно "качество", согласно серии стандартов ISO 9000, это ситуация, при которой потребители получают от производителя продукцию соответствующую их прямым требованиям и подспудным ожиданиям. Поэтому управление качеством, в соответствии с ISO 9000, предполагает применение т.н. "процессного подхода", когда моделируется и внедряется наиболее оптимальная цепь "преобразований-процессов", гарантирующая, что потребности потребителей воспринимаются производителем и воплощаются в любой продукт без искажений.

Такие понятия, как процессный подход, анализ и измерения, совершенствование процессов, введены в версиях ISO 9000 с 2000 года и ранее в них отсутствовали.

Следует заметить, что стандарты этой серии универсальны - они не ориентированы на какую либо конкретную отрасль, не учитывает специфики IT-сферы и, в этом смысле, конечно по степени конкретизации, заметно уступают, например, стандартам CMM.

Кроме того, ISO 9000 не предполагает никаких градаций (уровней) соответствия этим процессам, и, тем самым, затрудняет определение "истинных" возможностей той или иной организации и, соответственно, - путей их дальнейшего развития.

Карта процесса «Разработка и тестирование программного обеспечения»



Этап выполнения процесса	Ответственный	Результат
1. Анализ ТЗ на создание АСУ ТП и док-тов технического проекта системы. Определение требований к ПО, вкл. входные и выходные данные для разработки	Руководитель сектора ПО, ГИП	Спецификация входных и выходных данных для разработки ПО
2. Составление и утверждение плана разработки ПО по конкретному проекту с указанием сроков и исполнителей работ	Руководитель сектора ПО, ГИП, управляющий	Утвержденный план разработки ПО
3. Заимствование готовых программных компонентов из программного фонда	Руководитель сектора ПО, разработчик ПО	Программы и программная документация
4. Разработка недостающих программных компонентов ПО	Руководитель сектора ПО, разработчик ПО	Программы и программная документация
5. Разраб-ка программ тестовых испытаний и контрольных тестов	Разработчик ПО	Программы тестовых испытаний и контрольные тесты
6. Сборка, наладка и выполнение тестовых испытаний комплекта ПО. Несоответствия имеются?	Разработчик ПО	Перечень несоответствий
7. Оценка функциональности комплекта ПО (верификация). Комплект ПО соответствует требованиям ТЗ и ТНПА?	Руководитель сектора ПО, ГИП	Перечень несоответствий
8. Действия с несоответствующей пр-цией	В соответствии с СТП СМК 8.3.0-01 – 2009 «Управление несоответствующей продукцией»	
9. Утверждение результатов тестирования	Руководитель сектора ПО, ГИП	Решение о соответствии комплекта ПО требованиям
10. Разработка эксплуатационной документации по ПО	Разработчик ПО	Эксплуатационная документация в соотв. с ТЗ
11. Установка пакета ПО в сис-му. Проверка функционирования пакета ПО при комплексной отладке АСУ ТП (валидация)	Руководитель сектора ПО, ГИП	Перечень несоответствий
12. Анализ, выявление и устранение несоответствий ПО в процессе опыт. экспл-ции АСУ ТП. Корректировка программной док-ции	Разработчик ПО	Доработанные ПО и эксплуатационная док-ция по результатам валидации
13. Ввод ПО в промышленную экспл-цию и передача заказчику	Руководитель сектора ПО, ГИП	Акт ввода АСУ ТП в промышленную экспл-цию
14. Управление изменениями и сопровождение ПО	Руководитель сектора ПО, ГИП	Запросы на изменения. ПО с изменениями

Тема 2. Модели процесса разработки ПО

Тема 2. Модели процесса разработки ПО

■ CMMI (Capability Maturity Model Integration)

Это относительно новый стандарт в области менеджмента качества, который описывает модель зрелости процессов разработки программного обеспечения на предприятиях {3}.

Использование данной модели позволяет организации оценить эффективность бизнес-процессов, связанных с разработкой ПО, установить приоритетные направления их усовершенствования, а также внедрить данные усовершенствования.

Модель CMMI, а точнее ее версия 1.1, появилась в марте 2002 года. Она внедрена Институтом программной инженерии Университета Карнеги-Меллона (Software Engineering Institute, SEI) специально для отрасли разработки программного обеспечения.

До этого в конце 80-х годов для ПО по заказу Министерства обороны США была разработана SW-CMM.

Тема 2. Модели процесса разработки ПО

Основным компонентом модели СММІ является область процессов, определяющая цели и действия для их достижения. В качестве примера области процесса можно привести контроль качества процессов и продуктов (PPQA).

Другой областью процесса является управление конфигурацией (**СМ**). Важно понимать, что область процесса не является процессом. Один процесс может пересекаться с несколькими областями процесса, а одна область процесса может включать несколько процессов.

Область процессов (Process Area) – набор связанных практик данной области, исполняются для достижения ряда целей, которые считаются важными в контексте улучшения процессов в данной области.

В ходе оценки организации определяется уровень ее работы, который характеризует способность организации управлять рисками и, следовательно, выполнять взятые на себя обязательства.

Тема 2. Модели процесса разработки ПО

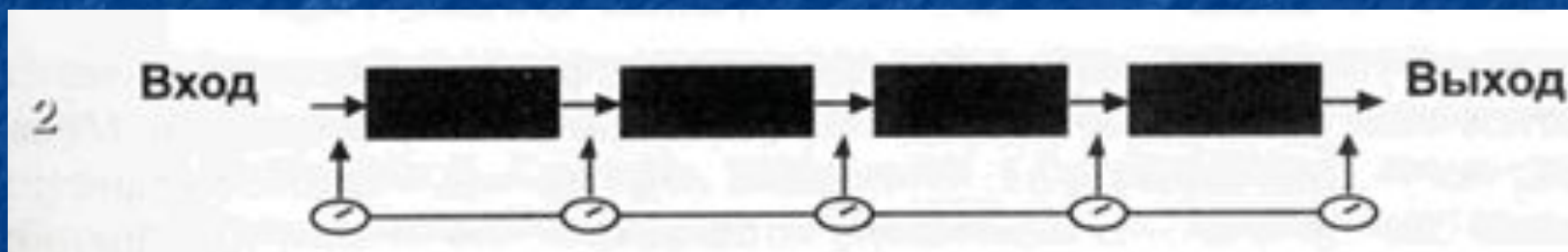
Процессы первого уровня зрелости, характеризуются хаотичностью, реактивностью, непредсказуемостью. Несмотря на это, очень часто организации, находящиеся на данном этапе развития, производят довольно качественные продукты. При этом, как правило, превышает бюджет и время разработки данных продуктов. Качественные продукты данных организаций производятся не за счет устойчивых и отлаженных процессов, а благодаря титаническим усилиям отдельных личностей. В случае ухода таких людей очень тяжело повторить успешные проекты. На данном этапе очень тяжело предсказать производительность процессов, протекающих в организации.



На уровне 1 производственный процесс (а вместе с ним и все процессы) представляется аморфной сущностью, практически черным ящиком, представление о процессах очень ограниченное, чрезмерно много усилий тратится на выяснение статуса развития проекта и текущего хода работ.

Тема 2. Модели процесса разработки ПО

Уровень зрелости 2 – управляемый уровень. На данном этапе основные процессы описаны, их, возможно, использовать неоднократно. Другими словами, проекты, выполняемые организацией, отвечают требованиям. Процессы управляемы, они планируются, выполняются, измеряются и контролируются. Однако процессы все же имеют некоторую долю реактивности в своей сущности.

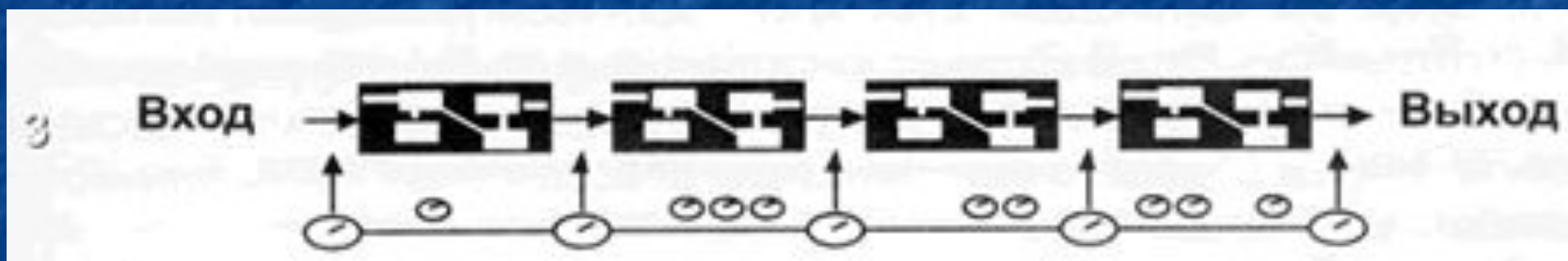


На уровне 2 контролируются требования заказчиков и промежуточные продукты, а также установлены основные практики управления проектом. Эти средства позволяют управлять проектом, однако дают фрагментарное представление о нем. Фактически, производственный процесс можно представить последовательностью черных ящиков и реальное видение проекта присутствует лишь на промежуточных этапах.

Тема 2. Модели процесса разработки ПО

Уровень зрелости 3 – определенный уровень. В этом случае процессы определены. Установлены стандарты в пределах организации. На данном этапе процессы описаны не на уровне отдельного проекта, а на уровне всей организации.

Присутствует более детальное описание всех процессов, в котором лучше раскрываются связи и зависимости, знание которых позволяет улучшить управление.



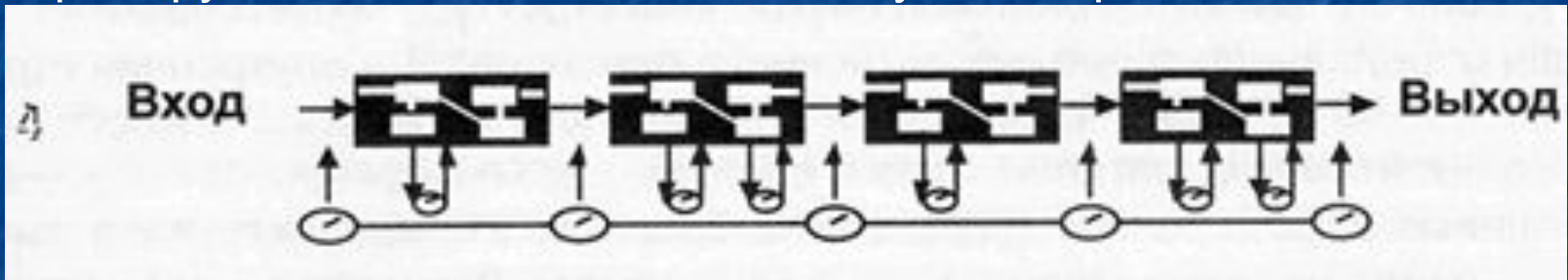
На этом уровне становится видимой внутренняя сторона наших черных ящиков. Это внутренняя структура отражает способ, применения стандартного производственного процесса организации.

Тема 2. Модели процесса разработки ПО

Уровень зрелости 4 – количественно-управляемый уровень.

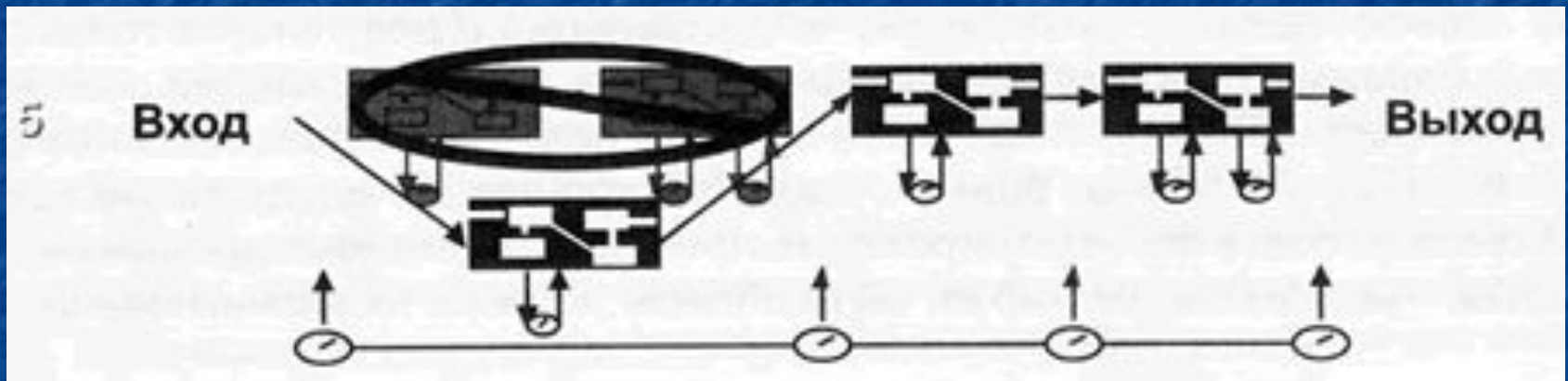
На данном этапе достигнуты все цели предыдущих уровней. Выбраны субпрактики, которые при использовании статистических методов и других количественных техник позволяют контролировать качество выполнения процессов. Самое главное отличие этого этапа от предыдущего заключается в предсказуемости эффективности процессов и возможности ею (эффективностью) управлять.

На уровне 4 определенные процессы количественно контролируются с помощью соответствующих средств и техник.



Тема 2. Модели процесса разработки ПО

Уровень зрелости 5 – уровень постоянного улучшения (оптимизации) процессов. На данном этапе мы имеем точные характеристики оценки эффективности бизнес процессов, что позволяет нам постоянно и эффективно улучшать бизнес процессы путем развития существующих методов и техник и внедрения новых.



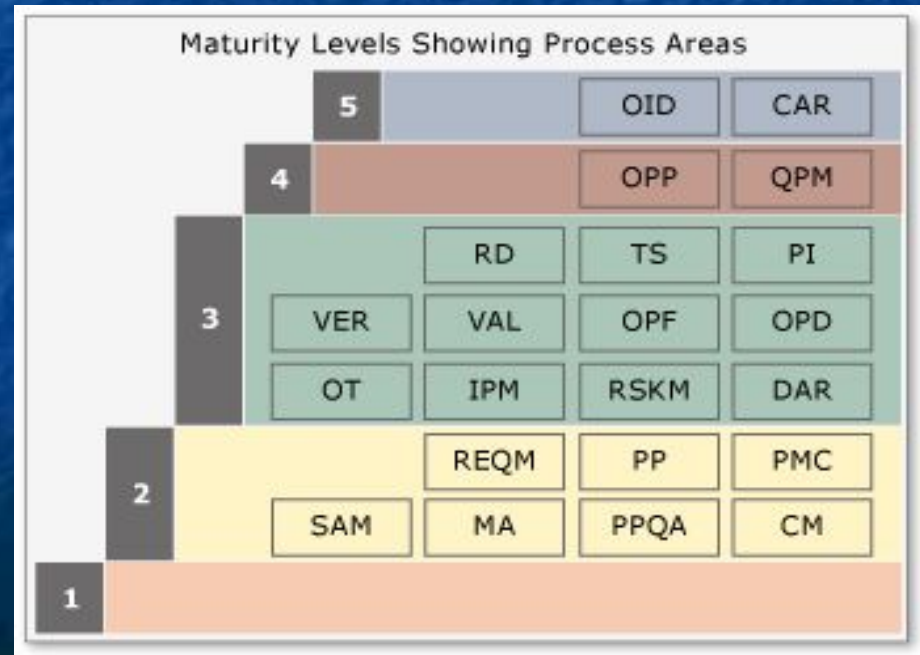
Акроним	Область процесса
CAR	Анализ и разрешение причин
CM	Управление конфигурацией
DAR	Анализ и разрешение решений
IPM	Интегрированное управление проектами
MA	Измерения и анализ
OID	Организационные инновации и развертывание
OPD	Определение организационных процессов
OPF	Фокус организационных процессов
OPP	Производительность организационных процессов
OT	Organizational Training
PI	Интеграция продуктов
PMC	Мониторинг и контроль проектов
PP	Планирование проектов
PPQA	Контроль качества процессов и продуктов
QPM	Количественное управление проектами
RD	Определение требований
REQM	Управление требованиями
RSKM	Управление рисками
SAM	Управление соглашениями с поставщиками
TS	Техническое решение
VER	Проверка
VAL	Проверка

Тема 2. Модели процесса разработки ПО

Уровень зрелости

Название уровня

- 1 Начальный уровень (**Initial**)
- 2 Управляемый уровень (**Managed**)
- 3 Определенный уровень (**Defined**)
- 4 Количественно-управляемый уровень (**Quantitatively Managed**)
- 5 Оптимизированный уровень (**Optimizing**)



Тема 2. Модели процесса разработки ПО

Уровни 4 и 5 называют уровнями высокой зрелости. Как правило, различия между зрелыми организациями, реализующими стратегии количественного управления и оптимизации, и незрелыми организациями с неэффективной системой управления и следованием заранее определенным процедурам, очевидны. Деятельность зрелых организаций отличается низкой вариативностью процессов и характеризуется использованием ключевых показателей в рамках реализации статистически оправданного метода управления. Следовательно, более зрелые организации отличаются большей предсказуемостью и более высокой скоростью реагирования на новую информацию (при отсутствии бюрократических механизмов, затормаживающих процесс).

Менее зрелые организации бросают на борьбу с экстренными ситуациями все силы, в то время как более зрелые слепо следуют устоявшимся процедурам и не всегда могут понять, что изменение некоторых процедур станет более адекватным ответом на сложившуюся ситуацию.

Уровень зрелости	Область процессов	Сокращение	Цель
Уровень 2	Менеджмент требований (Requirements Management)	REQM	Управление требованиями предъявляемым к продуктам проекта или компонентам продукта, с целью выявления несоответствия между требованиями и планами проекта.
	Планирование проекта (Project Planning)	PP	Разработка и поддержание планов определяющих развитие проекта
	Мониторинг и контроль проекта (Project Monitoring and Control)	PMC	Обеспечить понимание стадии разработки проекта с целью принятия корректирующих действий в случае серьезного отклонения от плана
	Менеджмент договоров с поставщиками (Supplier Agreement Management)	SAM	Управление приобретением товаров и услуг от внешних поставщиков, с которыми заключены договоры
	Измерение и анализ (Measurement and Analysis)	M&A	Разработка и поддержание возможности измерения, используемой для поддержки нужд информационного менеджмента
	Оценка (гарантирование) качества товаров и процессов (Process and Product Quality Assurance)	PPQA	Обеспечение поддержки и управления в соответствии с целями процессов и связанными с ними продуктами работы
	Конфигурационный менеджмент (Configuration Management)	CM	Установка и поддержание целостности продуктов работы (work products) в результате использования идентификации конфигураций, конфигурационного контроля и конфигурационного аудита
Уровень 3	Разработка требований (Requirements Development)	RD	Сбор и анализ требований потребителей к продуктам и компонентам продуктов
	Техническое решение (Technical Solution)	TS	Разработка, дизайн и внедрение решений по соответствующим требованиям. Решения, дизайн и внедрения выражены продуктами, компонентами продуктов и связанными с данными продуктами процессами.
	Интеграция продукта (Product Integration)	PI	Сборка (монтаж) продукта из его составляющих, проверка качества интеграции, ее функциональности и выпуск продукта.
	Верификация (Verification)	Ver	Гарантирование того, что выбранные продукты работы отвечают предъявляемым требованиям
	Валидация (Validation)	Val	Демонстрация того, что продукт и его компоненты соответствуют его предполагаемому использованию в предполагаемой среде.
	Фокусирование на процессах организации (Organization Process Focus)	OPF	Установление и поддержание понимания процессов организации и процессных активов, идентификация, планирование и внедрение улучшений связанных с данными областями.
	Описание процессов организации (Organization Process Definition)	OPD	Установление и поддержание возможного к использованию массива процессов организации
	Организационный тренинг (Organizational Training)	OT	Повышение знаний и способностей людей для выполнения ими своих ролей эффективно и рационально
	Менеджмент интеграции проектов (Integrated Project Management)	IPM	Установка и управление проектом и вовлечение всех заинтересованных лиц в интегрированный и определенный процесс. Данная область также затрагивает общее видение проекта командой разработчиков
	Менеджмент рисков (Risk Management)	RSKM	Определение потенциальных проблем до их появления. В связи с этим процессы по снижению рисков могут планироваться и осуществляться на любом этапе разработки продукта или процесса.
	Интегрированные команды (разработчиков) Integrated Teaming	IT	Формирование и поддержание интегрированных команд для разработки продуктов работы (work products)
	Интегрированное управление поставщиками (Integrated Supplier Management)	ISM	Мониторинг новых продуктов, оценка источников продуктов, которые могут удовлетворить требованиям к проекту и использование данной информации для выбора поставщиков
	Анализ решений и разрешение (Decision Analysis and Resolution)	DAR	Разработка решений на основе структурированного подхода, который позволяет оценить альтернативные решения на основе установленных критериев
	Организационная среда для интеграции (Organizational Environment for Integration)	OEI	Предоставление инфраструктуры для интегрированной разработки продуктов и процессов и управление людьми (персоналом) в целях интеграции
	Уровень 4	Производительный организационный процесс (Organizational Process Performance)	OPP
Количественный менеджмент проекта (Quantitative Project Management)		QPM	Количественно управлять определенным процессом в целях достижения установленного в рамках проекта качества и целей производительности.
Уровень 5	Организационные инновации и внедрение (Organizational Innovation and Deployment)	OID	Выбор и внедрение инноваций и улучшений, которые измеряемо, улучшают организационные процессы и технологии.
	Анализ причин и разрешение (Causal Analysis and Resolution)	CAR	Идентификация причин дефектов и других проблем и принятие действий предотвращающих их появление в будущем

Тема 2. Модели процесса разработки ПО

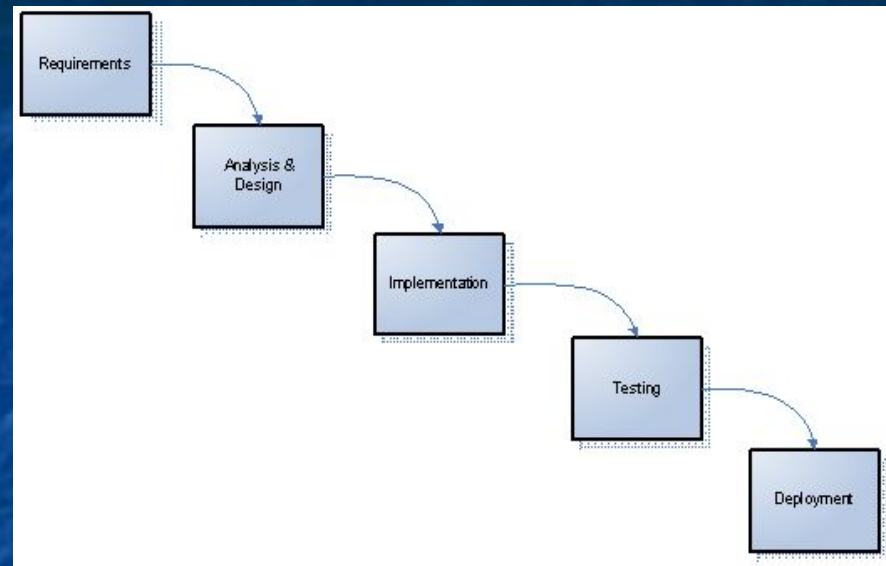
Центр программного производства EPAM Systems в Минске в конце сентября 2003 г. первым в Европе успешно прошел международную аттестацию соответствия 4-му уровню CMMI® (SEI CMMI Maturity Level 4).

Группа компаний IBA 2003 г. прошла международную аттестацию соответствия 4-му уровню CMMI® (SEI CMMI Maturity Level 4).

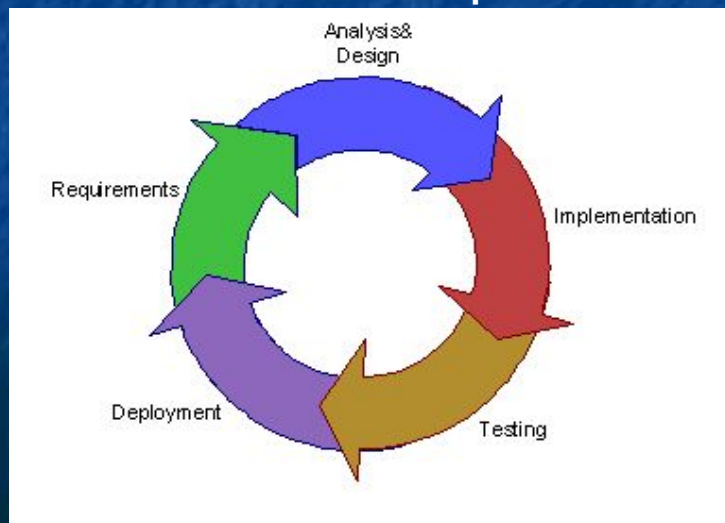
Тема 2. Модели процесса разработки ПО

Для успешного выполнения программного проекта недостаточно выбрать эффективные средства разработки, обеспечить необходимый бюджет и найти квалифицированных разработчиков. В любой организации существуют правила и методики, по которым участники проекта (заказчики, разработчики, тестеры, технические писатели) распределяют между собой задачи, взаимодействуют друг с другом, создают проектные артефакты (спецификации, исходный код, документацию). Эти правила могут быть четко организованными или хаотичными, быть формально документированными или существовать в головах проектной команды, но в любом случае именно их совокупность называется **процессом разработки**.

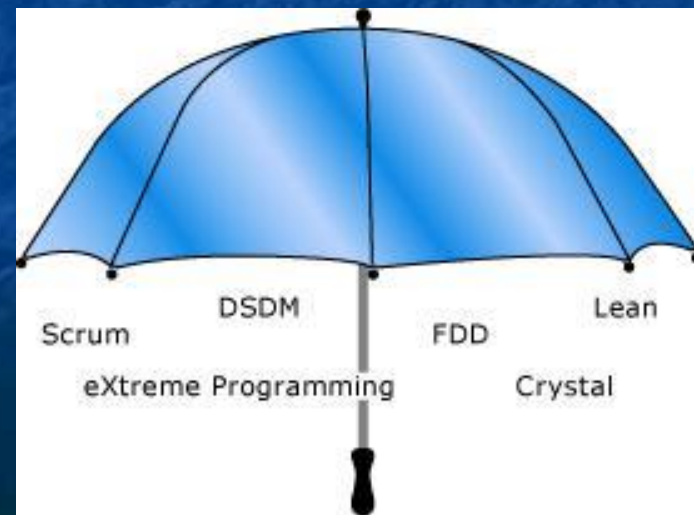
Модель водопада



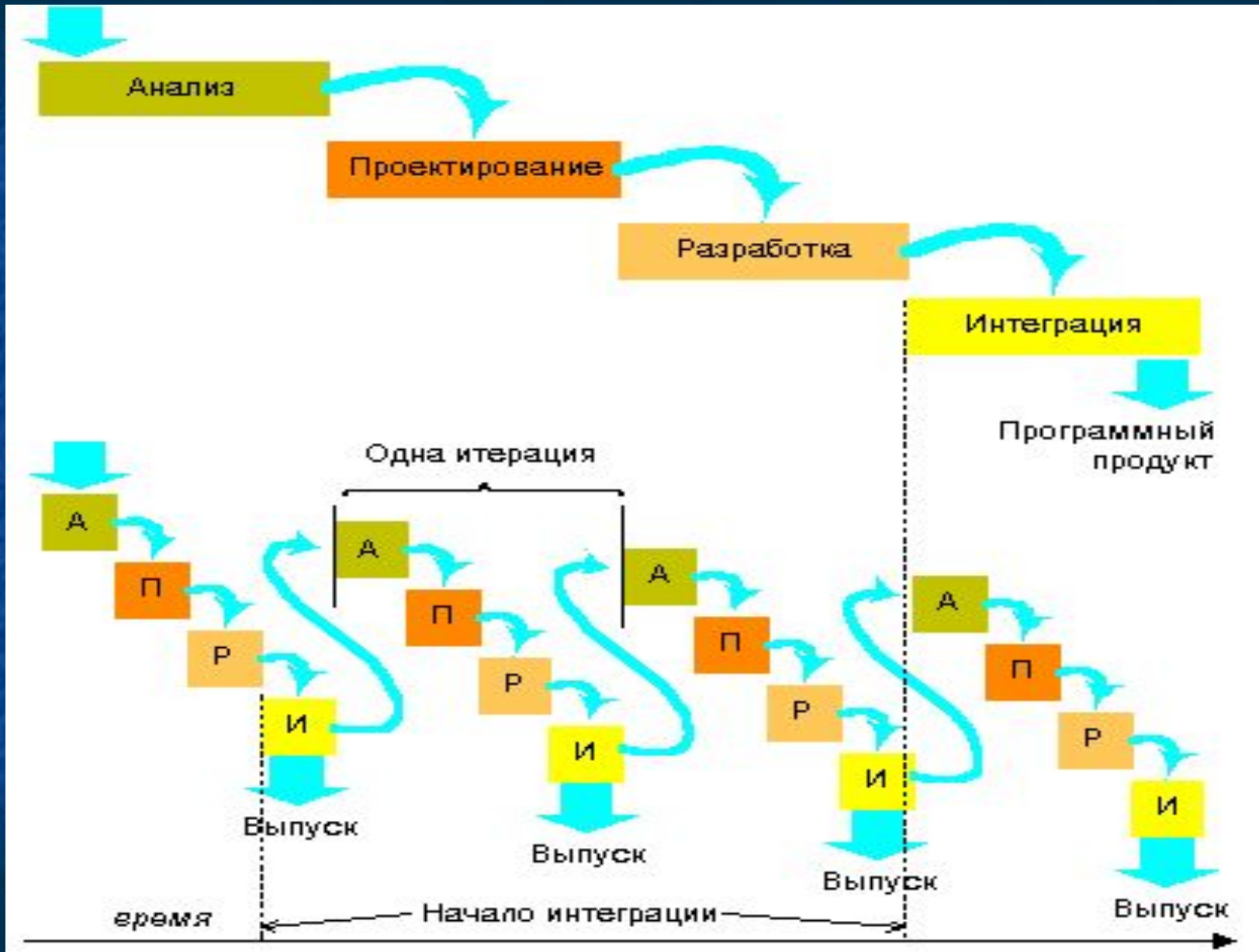
Итерационная инкрементальная модель The Unified Software Development Process



Гибкие методологии разработки ПО (Agile)



Тема 2. Модели процесса разработки ПО



Тема 2. Модели процесса разработки ПО

Метод водопада

Модель водопада (waterfall model или последовательная разработка) – самый известный, исторически появившийся одним из первых процесс разработки. Основная идея заключается в том, что процесс разработки делится на четко определенные фазы, выполняемые строго последовательно.

Фазы проекта включают следующие этапы:

- **Разработка требований** (requirements): сбор бизнес-требований заказчика и их преобразование в функциональные требования к программному продукту.
- **Анализ и дизайн** (analysis and design): разработка модели предметной области (domain model), проектирование схемы базы данных, объектной модели, пользовательского интерфейса и т.п.
- **Реализация** (implementation): создание продукта по спецификациям, разработанным на предыдущем этапе.
- **Тестирование** (testing): включает проверку соответствия функциональности программного продукта потребностям пользователей (validation), а также поиск дефектов в реализации.
- **Развертывание** (deployment): обучение пользователей, инсталляция системы, перевод в промышленную эксплуатацию.

Недостатки модели

1. Процесс плохо работает в проектах с нечеткими требованиями. Даже если проектная команда считает, что полностью проработала и документировала функциональный дизайн системы, он может значительно отличаться от ожиданий пользователей. С большой вероятностью это расхождение будет обнаружено не на этапе рецензирования функциональной спецификации (редкий заказчик способен представить поведение реальной системы, читая документ с описанием ее функциональности), а во время внедрения продукта.
2. Процесс крайне неэффективен при постоянных изменениях требований (что как правило случается в проектах, длящихся больше одного-двух месяцев). Каждое изменение заставляет возвращаться к фазе определения требований и повторять весь процесс с начала.

Тема 2. Модели процесса разработки ПО

3. Сложно управлять рисками некоторых типов (таких, как риски, связанные с использованием новых технологий или риски некорректного определения требований). Подобные риски могут проявить себя только на этапе реализации (если не тестирования), когда число возможных путей исправления ситуации намного меньше, чем в начале проекта.
4. Весьма ограничены возможности оценки и корректировки важных атрибутов проекта – скорости разработки, качества продукта, обоснованности принятых архитектурных решений. Адекватно оценить эти атрибуты становится возможным только на поздних этапах проекта.

Итеративная разработка

Процесс итеративной (или инкрементальной) разработки стал эволюционным развитием модели водопада. Процесс состоит из серии повторяющихся итераций (их число зависит от конкретного проекта), каждая из которых фактически является полноценным мини-проектом с фазами определения требований, анализа, дизайна и т.д.

В результате очередной итерации продукт приобретает новую функциональность или улучшения в существующей функциональности. Полный набор требований, зафиксированный границами проекта, оказывается реализованным после завершения финальной итерации.

RUP (Rational Unified Process)

Один из самых известных процессов, использующих итеративную модель разработки – Rational Unified Process (RUP). Он был создан во второй половине 1990-х годов в компании Rational Software. Основными разработчиками были Филипп Крачтен (Philippe Kruchten), Грейди Буч (Grady Booch), Джеймс Рамбо (James Rumbaugh) и Айвар Якобсон (Ivar Jacobson). Кстати, последние трое являются также создателями нотации UML.

Термин RUP означает как методологию разработки, так и продукт компании IBM (ранее – Rational) для управления процессами разработки. Методология RUP описывает абстрактный общий процесс, на основе которого организация или проектная команда должна создать специализированный процесс, ориентированный на ее потребности.

Тема 2. Модели процесса разработки ПО

Рабочий процесс RUP

В терминах RUP участники проектной команды создают так называемые артефакты (work products), выполняя задачи (tasks) в рамках определенных ролей (roles). Артефактами являются спецификации, модели, исходный код и т.п.

Задачи разделяются по девяти процессным областям, называемым дисциплинами (discipline). В RUP определены шесть инженерных и три вспомогательные дисциплины.

В них входят:

Бизнес-моделирование (*Business Modeling*) – определение и описание бизнес-целей, моделирование сущностей предметной области (сущности и их атрибуты, структура связей, взаимодействие внешних и внутренних сущностей), моделирование функциональных обязанностей бизнес-актеров, анализ и моделирование существующих бизнес-процессов заказчика, а также поиск их возможных улучшений.

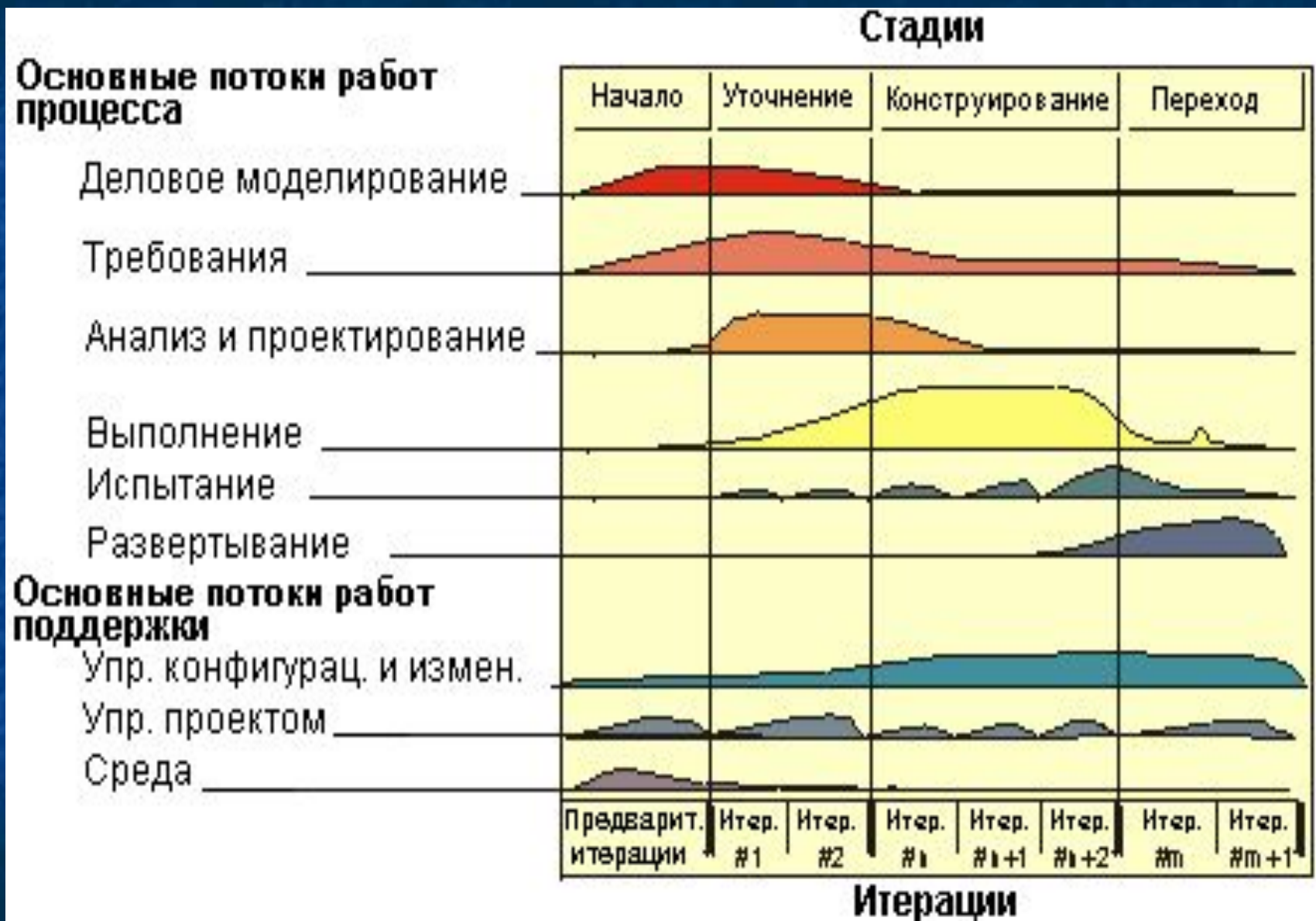
Разработка и Управление требованиями (*Requirements Engineering & Requirements Management*) – выявление требований, определение границ проекта, разработка функционального дизайна будущей системы и его согласование с заказчиком.

Анализ и проектирование (*Analysis and Design*) – проектирование архитектуры системы на основе функциональных требований и ее развитие на протяжении всего проекта.

Анализ (*Analysis*): выявление объектов программной системы, отвечающих за ее бизнес-логику, путем анализа потоков событий, описанных в сценариях, моделирование их взаимодействия и структуры связей, моделирование классов бизнес-логики, их иерархий и структуры связей (платформонезависимая модель PIM).

Проектирование (*Design*): создание и моделирование платформозависимой архитектуры программной системы (модель PSM).

Тема 2. Модели процесса разработки ПО



Тема 2. Модели процесса разработки ПО

Реализация (*Implementation*) – разработка, юнит-тестирование и интеграция компонентов системы.

Тестирование (*Test*) – поиск и отслеживание дефектов в системе, проверка корректности реализации требований.

Развертывание (*Deployment*) – создание дистрибутива, установка системы, обучение пользователей.

Управление конфигурациями и изменениями (*Configuration and Change Management*) – управление версиями исходного кода и документации, процесс обработки запросов на изменение (*change requests*).

Управление проектом (*Project Management*) – создание проектной команды, планирование фаз и итераций, управление бюджетом и рисками.

Среда (*Environment*) – создание инфраструктуры для выполнения проекта, включая организацию и настройку процесса разработки.

Фазы проекта:

Начало (*Inception*)

Цели:

- понять границы проекта;
 - разработка экономического обоснования;
 - заключение соглашения между заинтересованными сторонами для дальнейшего продвижения
- Веха: Разработка целей жизненного цикла

Проектирование (*Elaboration*)

Цели:

- Свести к минимуму главные технические риски;
 - Создать базовую архитектуру;
 - Понять во что обойдется построение системы
- Веха: архитектура жизненного цикла

Построение (*Construction*)

Цели:

Построить первую работающую версию продукта
Веха: начальная функциональная готовность

Внедрение (*Transition*)

Цели:

Создать окончательную версию программного продукта и передать ее Заказчику
Веха: готовый продукт

Тема 2. Модели процесса разработки ПО

Гибкие методологии разработки ПО (Agile)

Гибкая разработка — это термин, произошедший от так называемого "Гибкого манифеста" (Agile Manifesto), который был написан в 2001 г. группой, включавшей в себя создателей Scrum, экстремального программирования (XP), метода разработки динамических систем (DSDM) и Crystal; представителя разработки, управляемой функциями; а также некоторых других лидеров мысли из отрасли программного обеспечения.

"Гибкий манифест" установил общий набор всеохватывающих ценностей и принципов для всех отдельных гибких методологий, существовавших на то время. В нем детализированы четыре ключевых ценности, позволяющих командам достигать высокой производительности.

Agile-манифест разработки программного обеспечения

Мы постоянно открываем для себя более совершенные методы разработки программного обеспечения, занимаясь разработкой непосредственно и помогая в этом другим. Благодаря проделанной работе мы смогли осознать, что:

- *Люди и взаимодействие важнее процессов и инструментов*
- *Работающий продукт важнее исчерпывающей документации*
- *Сотрудничество с заказчиком важнее согласования условий контракта*
- *Готовность к изменениям важнее следования первоначальному плану*

То есть, не отрицая важности того, что справа, мы всё таки больше ценим то, что слева.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Основопологающие принципы Agile-манифеста

Мы следуем таким принципам:

Наивысшим приоритетом для нас является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке ценного программного обеспечения.

Изменение требований приветствуется, даже на поздних стадиях разработки. Agile-процессы позволяют использовать изменения для обеспечения заказчику конкурентного преимущества.

Рботающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев.

На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе.

Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им.

Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды.

Рботающий продукт — основной показатель прогресса.

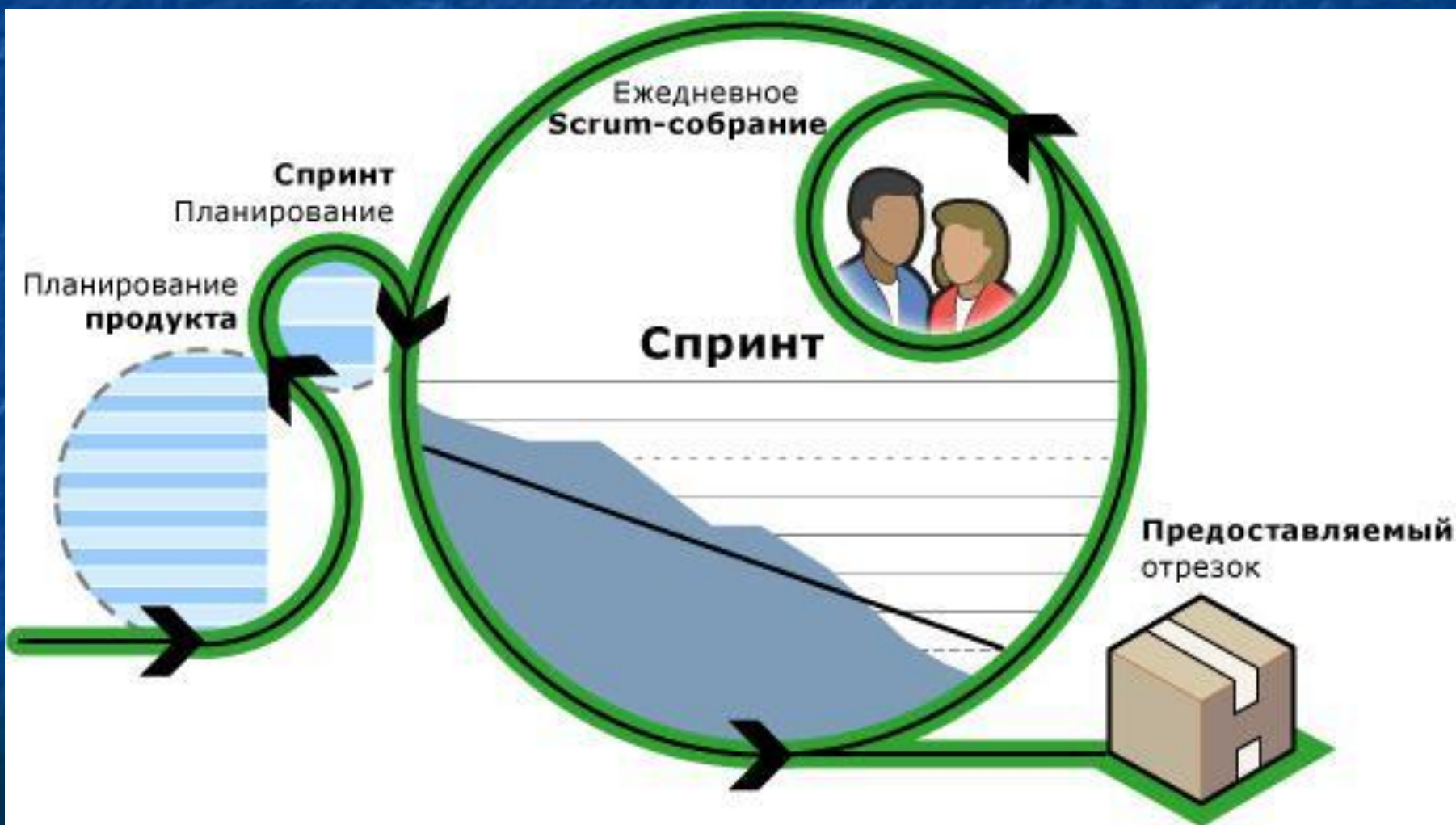
Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно. Agile помогает наладить такой устойчивый процесс разработки.

Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта.

Простота — искусство минимизации лишней работы — крайне необходима.

Самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд.

Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.



ТЕМА 3
МЕТОДЫ УПРАВЛЕНИЯ
ПРОЕКТАМИ

Тема 3. Методы управления проектами

В качестве самостоятельной области знаний управление проектами начало формироваться в начале XX века. В этой дисциплине пока нет единых международных стандартов.

Наиболее известные центры компетенции:

- PMI, Project Management Institute, PMBOK — американский национальный стандарт ANSI/PMI 99-001-2004.
- IPMA, International Project Management Association. В России — СОВНЕТ.

Тема 3. Методы управления проектами

ОСНОВНЫЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

- **Проект (Project)** – уникальный комплекс взаимосвязанных мероприятий, направленных на достижение конкретной цели при определенных требованиях к срокам, бюджету и характеристикам ожидаемых результатов.
- **Процесс управления (Management process)** описывает действия (работы - activities), предпринимаемые для обеспечения того, что процессы программной инженерии выполняются в согласовании с политиками, целями и стандартами, принятыми в организации
- **Управление проектом (Project Management)** — это процесс планирования, организации и контроля за состоянием задач и ресурсов проекта, направленный на своевременное достижение цели проекта при балансировании между объемом работ, ресурсами (такими как деньги, труд, материалы, энергия, пространство и др.), временем, качеством при балансировании между объемом работ, ресурсами (такими как деньги, труд, материалы, энергия, пространство и др.), временем, качеством и рисками.
- **Масштаб проекта** (иногда этот термин заменяют словосочетанием «содержание и границы проекта») — это совокупность цели проекта и планируемых для ее достижения аспектов времени и средств

Тема 3. Методы управления проектами

Тройственная Ограниченность

В основе современных методов управления проектами лежат методики структуризации работ и сетевого планирования, разработанные в конце 50-х годов XX века в США

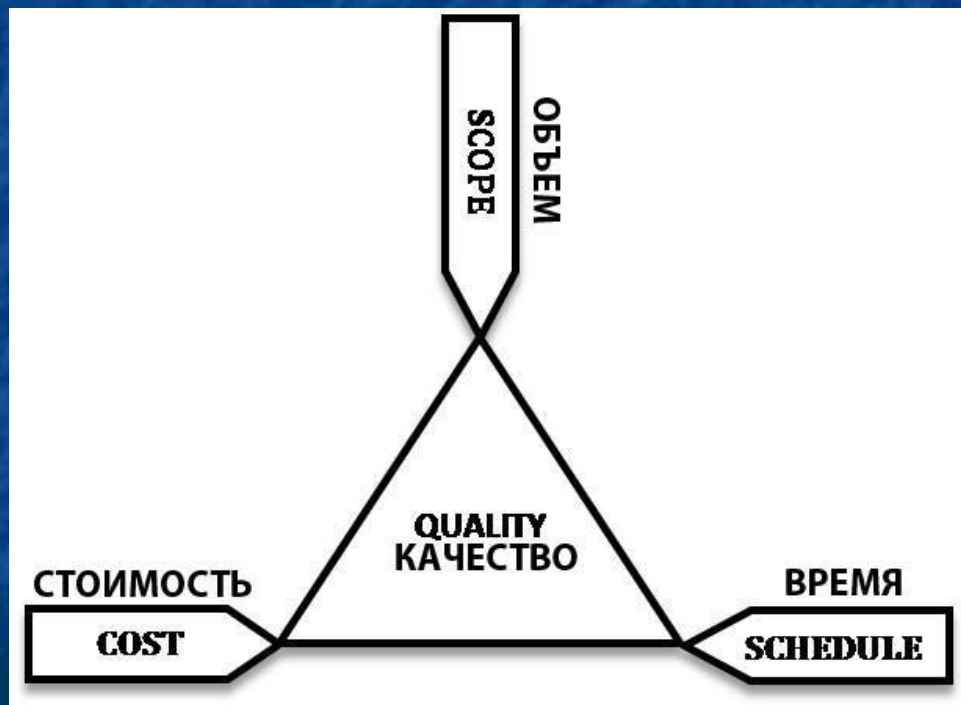


Рис.1. Треугольник управления проектами

Классическая форма Тройственной Ограниченности описывает баланс между содержанием, стоимостью, временем и качеством -- ограничения, в рамках которых каждый проект должен протекать и достигать финала.

Изменение одной стороны треугольника влияет на другие стороны.

Тема 3. Методы управления проектами

Тройственная Ограниченность

Ограниченность времени определяется количеством доступного времени для завершения проекта.

Ограниченность стоимости определяется бюджетом, выделенным для осуществления проекта.

Ограниченность содержания определяется набором действий, необходимых для достижения конечного результата проекта.

Эти три ограниченности часто соперничают между собой. Изменение содержания проекта обычно приводит к изменению сроков (времени) и стоимости:

- сжатые сроки (время) могут вызвать увеличение стоимости и уменьшение содержания.
- небольшой бюджет (стоимость) может вызвать увеличение сроков (времени) и уменьшение содержания.

Тема 3. Методы управления проектами

Управление проектами является наукой о применении инструментов и технологий, которые дают возможность команде (не только управляющему проектом) организовать работу с учётом этих ограничений.

Иной подход к управлению проектами рассматривает следующие три ограниченности: **финансы, время и человеческие ресурсы.**

При необходимости сократить сроки (время) можно увеличить количество занятых для решения проблемы людей, что непременно приведет к увеличению бюджета (стоимость). За счет того, что эта задача будет решаться быстрее, можно избежать роста бюджета, уменьшая затраты на равную величину в любом другом сегменте проекта.

Тема 3. Методы управления проектами

Подход **Предположение о неограниченности ресурсов**, критичен только срок выполнения. Метод PERT, критичен только срок выполнения. Метод PERT, Метод критического, критичен только срок выполнения. Метод PERT, Метод критического пути, критичен только срок выполнения. Метод PERT, Метод критического пути;

- **Предположение о критичности качества** (под качеством здесь понимается полнота удовлетворения требований, как известных, так и неизвестных заранее) Гибкая методология разработки (Agility);
- **Предположение о неизменности требований и низких рисках.** Классические методы (например, RUP, спиральная модель, во многом опирающиеся на модель водопада);
- **Предположение о высоких рисках проекта.** Метод Инновационные проекты (стартапы)
- Варианты нейтральных (сбалансированных) подходов:
 - Акцент на взаимодействие исполнителей. Метод PRINCE2
 - Акцент на взаимодействие процессов. Метод Process-based management

Тема 3. Методы управления проектами

Методы

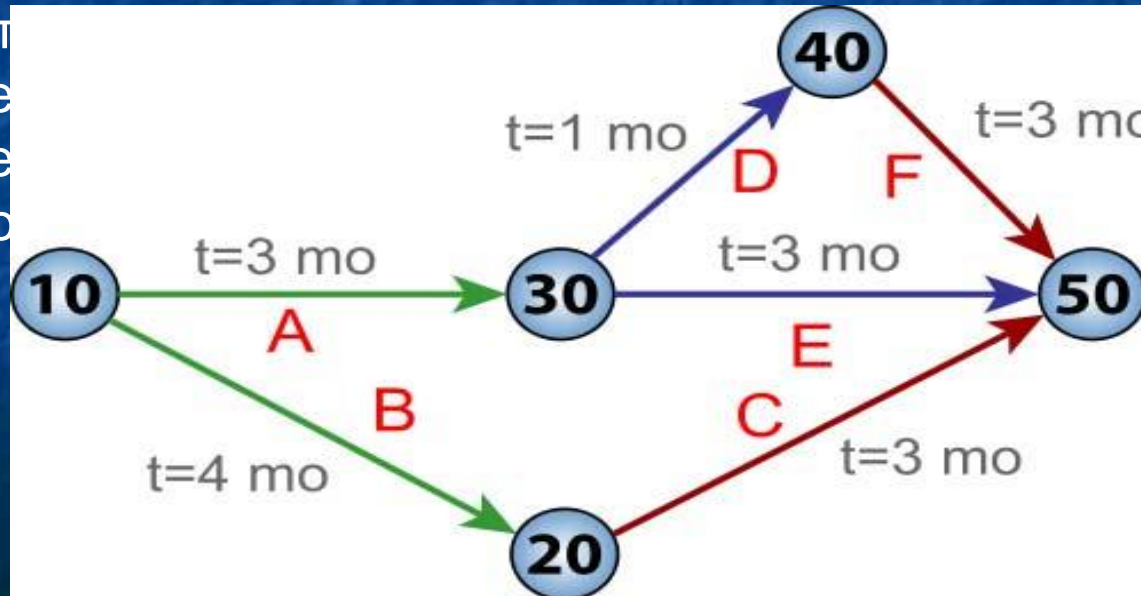
- **Program (Project) Evaluation and Review Technique** (сокращенно PERT) — техника оценки и анализа [программ](#)) — техника оценки и анализа программ ([проектов](#)), в особенности, анализа времени, которое требуется для выполнения каждой отдельной задачи, а также определение минимального необходимого времени для выполнения всего проекта.
- Самая известная часть PERT — это диаграммы взаимосвязей работ и событий. Предлагает использовать диаграммы-[графы](#) Самая известная часть PERT — это диаграммы взаимосвязей работ и событий. Предлагает использовать диаграммы-графы с работами на узлах, с работами на стрелках ([сетевые графики](#) Самая известная часть PERT — это диаграммы взаимосвязей работ и событий. Предлагает использовать диаграммы-графы с работами на узлах, с работами на стрелках (сетевые графики), а также [диаграммы Ганта](#).

Тема 3. Методы управления проектами

Диаграмма PERT с работами на стрелках представляет собой множество точек-вершин (события) вместе с соединяющими их ориентированными дугами (работы). Всякой дуге, рассматриваемой в качестве какой-то работы из числа нужных для осуществления проекта, приписываются определенные количественные характеристики. Это — объёмы выделяемых на данную работу ресурсов и, соответственно, её ожидаемая продолжительность (длина дуги). Любая вершина интерпретируется как событие завершения работ, представленных дугами, которые входят в неё, и одновременно начала работ, отображаемых дугами, исходящими оттуда. Таким образом отражается тот факт, что ни к одной из работ нельзя приступить прежде, чем будут выполнены все работы, предшествующие ей согласно технологии реализации проекта. Начало этого процесса — вершина без входящих, а окончание — вершина без исходящих дуг. Остальные вершины должны иметь и те, и другие дуги.

Тема 3. Методы управления проектами

Последовательность дуг, в которой конец каждой предшествующей совпадает с началом последующей, трактуется как путь от отправной вершины к завершающей, а сумма длин таких дуг — как его продолжительность. Обычно начало и конец реализации проекта Последовательность дуг, в которой конец каждой предшествующей совпадает с началом последующей, трактуется как путь от отправной вершины к завершающей, а сумма длин таких дуг — как его продолжительность. Обычно начало и конец реализации проекта связаны множеством путей, длины которых различаются. Наибольшая определяет длительность всего этого проекта, минимально возможную при зафиксированных характеристиках дуг графа. Соот



нно от
ая
ительности
ть.

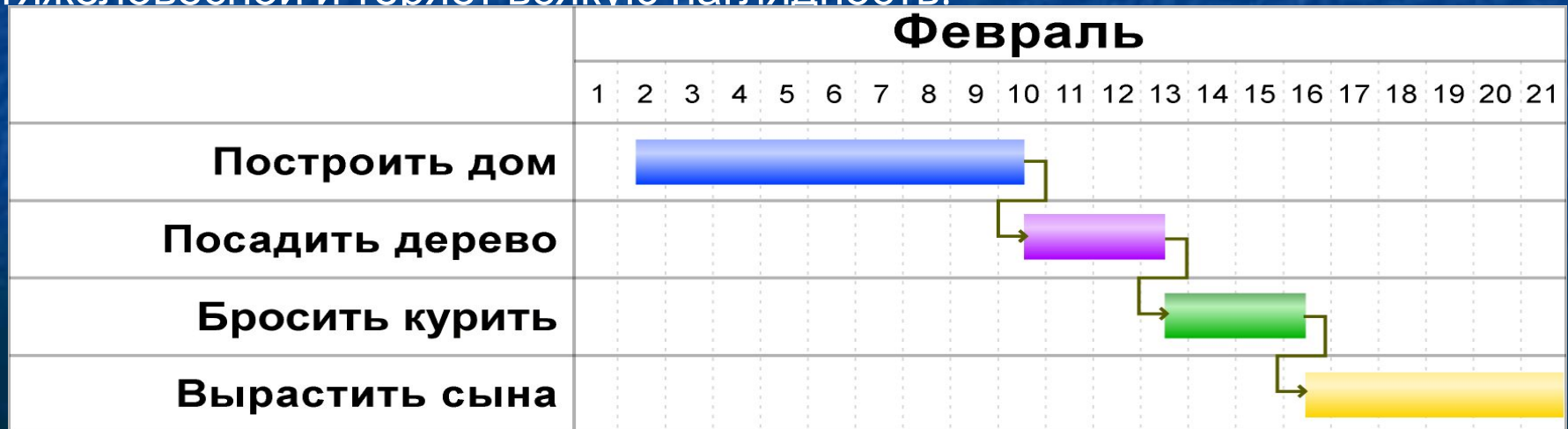
продолжите
продолжите
любых рабо

Тема 3. Методы управления проектами

- **Диаграмма Ганта** ([англ. Gantt chart](#), также **ленточная диаграмма, график Ганта**) — это популярный тип столбчатых [диаграмм](#)) — это популярный тип столбчатых диаграмм ([гистограмм](#)) — это популярный тип столбчатых диаграмм (гистограмм), который используется для иллюстрации плана, графика работ по какому-либо [проекту](#)) — это популярный тип столбчатых диаграмм (гистограмм), который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Используется в приложениях по [управлению проектами](#).
- Первый формат диаграммы был разработан [Генри Л. Гантом](#) в 1910 году.
- По сути, диаграмма Ганта состоит из полос, ориентированных вдоль оси времени. Каждая полоса на диаграмме представляет отдельную задачу в составе проекта (вид работы), её концы — моменты начала и завершения работы, её протяженность — длительность работы. Вертикальной осью диаграммы служит перечень задач. Кроме того, на диаграмме могут быть отмечены совокупные задачи, проценты завершения, указатели

Тема 3. Методы управления проектами

■ Ключевым понятием диаграммы Ганта является «Веха» — метка значимого момента в ходе выполнения работ, общая граница двух или более задач. Вехи позволяют наглядно отобразить необходимость синхронизации, последовательности в выполнении различных работ. Вехи, как и другие границы на диаграмме, не являются календарными датами. Сдвиг вехи приводит к сдвигу всего проекта. Поэтому диаграмма Ганта не является, строго говоря, графиком работ. И это один из основных её недостатков. Кроме того, диаграмма Ганта не отображает значимости или ресурсоемкости работ, не отображает сущности работ (области действия). Для крупных проектов диаграмма Ганта становится чрезмерно тяжеловесной и теряет всякую наглядность.



ТЕМА 4
ИНИЦИАЦИЯ
ПРОЕКТА

Тема 4. Инициация проекта

Описание проблемы

Формулировка концепции

- Разработка концепции (Vision);
- Границы системы;
- Ключевые пользовательские функции (User Requirements или Business Use Cases);
- Экономическое обоснование (Business Case):
- Является ли проект прибыльным?
- Является ли он реальным?

Предложить возможное решение (архитектуру)

Перечень рисков

Основные участники

План работ

Смета работ

Даты главных вех

1. Веха целей жизненного цикла ЦЖЦ. Конец фазы «НАЧАЛО». Четко определены границы проекта и финансирование;
2. Веха архитектуры жизненного цикла АЖЦ. Конец фазы ПРОЕКТИРОВАНИЯ. Завешена архитектура, заложена основа набора требований.
3. Веха начальной функциональной готовности НФГ. Конец фазы ПОСТРОЕНИЕ. Первый бетта-релиз.
4. Веха готового продукта ГП. Конец фазы ВНЕДРЕНИЕ и всего цикла разработки.