

ПМЗ

# МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ ПО

Лекция 4-2. Сравнение методологий разработки программ по степени формальности и отношению к каскадной или итеративной разработке.

Описание ГОСТов 19-й и 34-й серий и ИСО МЭК 12207, моделей зрелости процесса разработки (СММ, СММІ) по этим же параметрам

# Рассмотренные технологические подходы

## 1. Подходы со слабой формализацией

## 2. Строгие (классические, жесткие, предсказуемые) подходы

- 2.1. Каскадные технологические подходы.
  - 2.1.1. Классический каскадный подход
  - 2.1.2. Каскадно-возвратный подход
  - 2.1.3. Каскадно-итерационный подход
  - 2.1.4. Каскадный подход с перекрывающимися процессами
  - 2.1.5. Каскадный подход с подпроцессами
  - 2.1.6. Спиральная модель
- 2.2. Каркасные подходы.
  - 2.2.1. Рациональный унифицированный процесс (RUP)
- 2.3. Генетические подходы.
  - 2.3.1. Синтезирующее программирование
  - 2.3.2. Сборочное (расширяемое) программирование
  - 2.3.3. Конкретизирующее программирование
- 2.4. Подходы на основе формальных преобразований.
  - 2.4.1. Технология стерильного цеха
  - 2.4.2. Формальные генетические подходы

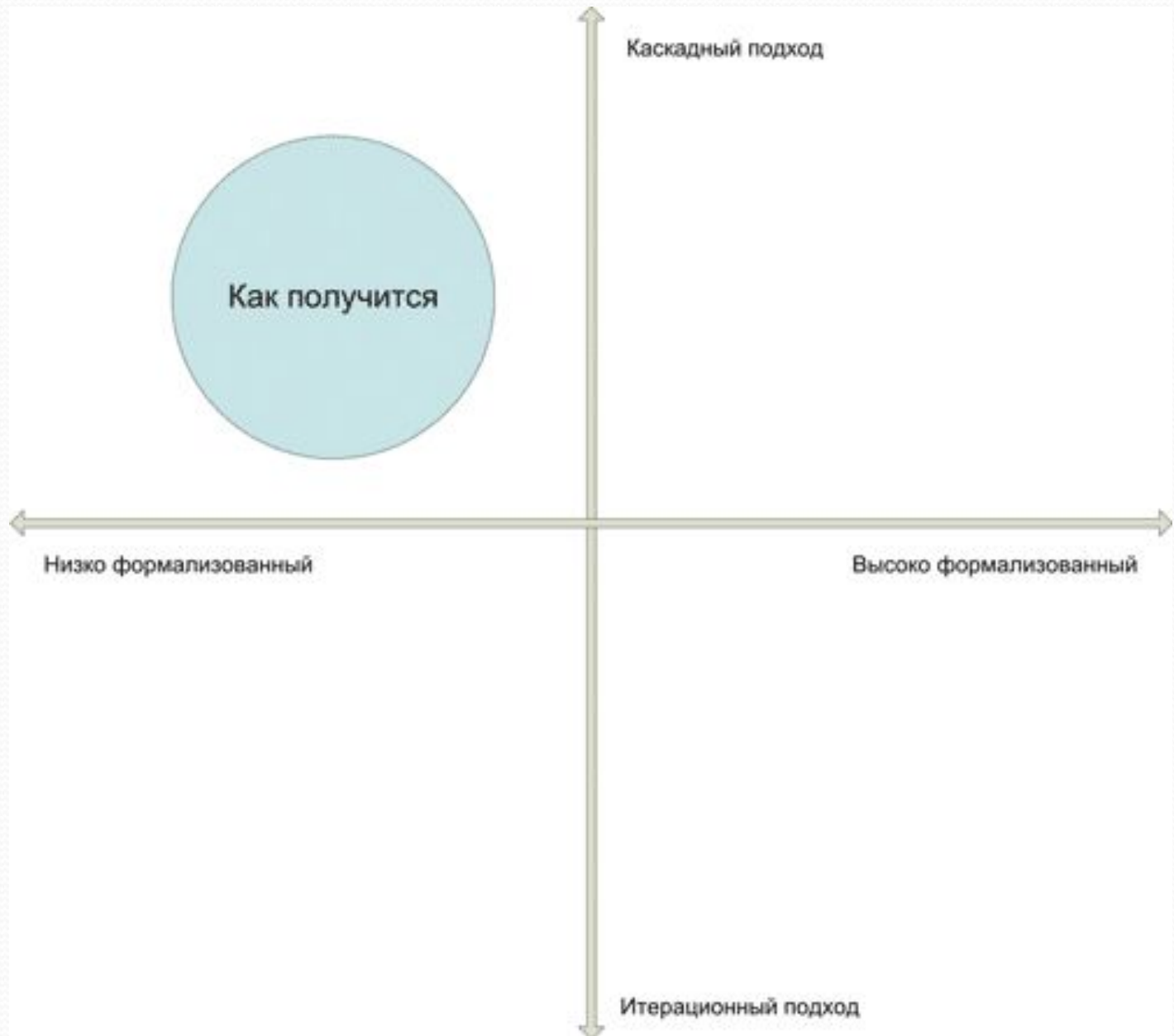
## ● 3 Гибкие (адаптивные, легкие) подходы

- 3.1. Ранние технологические подходы быстрой разработки (RAD)
  - 3.1.1. Эволюционное прототипирование
  - 3.1.2. Итеративная разработка
  - 3.1.3. Постадийная разработка
- 3.2. Адаптивные подходы.
  - 3.2.1. Экстремальное программирование (XP)
  - 3.2.2. Адаптивная разработка
- 3.3. Подходы исследовательского программирования.
  - 3.3.1. Компьютерный дарвинизм

# Сравнение основных методологий разработки ПО по двум показателям:

- 1) отношение к каскадной или итеративной разработке
- 2) степень формальности

# Методология «Как получится»



# Методология «Как получится»

Чаще всего это означает, что правил ведения разработки

- либо не существует вообще,
- либо они разработаны и приняты, но не выполняются.

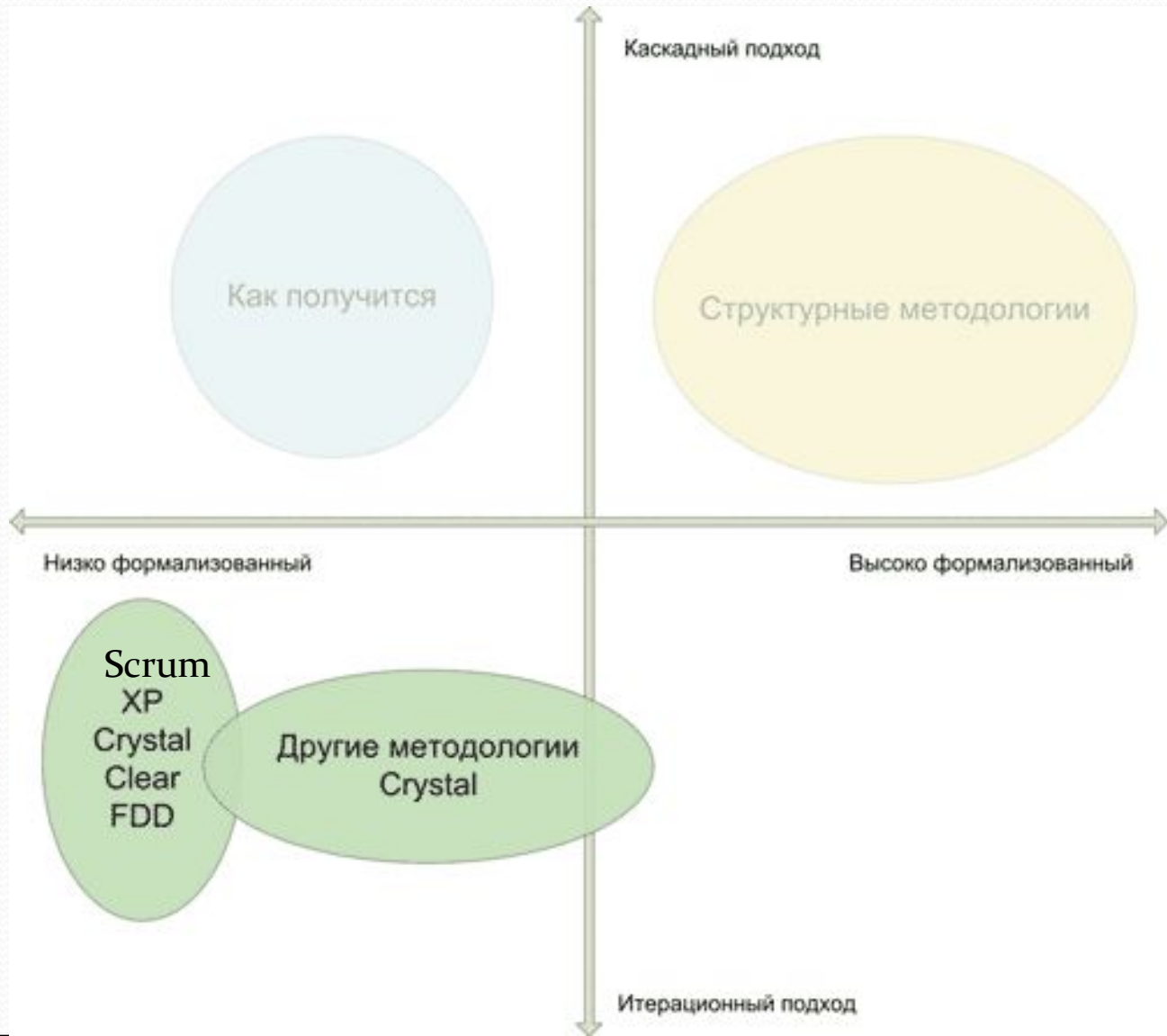
Естественным в таких условиях является крайне низкий уровень формализма разработки.

- **СТРУКТУРНЫЕ МЕТОДЫ** — это группа методологий, разработанных, как правило, еще до широкого распространения объектно-ориентированных языков. Все они предполагают каскадную разработку. Часто цитируют, что желательно начинать (в каскадном подходе) проект с разработки прототипа, то есть выполнять как минимум две итерации.
- Тем не менее основу этих методологий составляют последовательный переход от работы к работе и передача результатов (документов) очередного этапа участникам последующего.
- Также все эти методологии предполагают высокоформализованный подход, хотя высказывания о разумном количестве документации можно найти и в них.

# Структурные методологии



# Гибкие методологии





Назовем принципы, определяющие оценку этих (гибких) методологий по выбранным параметрам:

- главное — удовлетворить заказчика и предоставить ему продукт как можно скорее;
  - новые выпуски продукта должны появляться раз в несколько недель, в крайнем случае — месяцев;
  - наиболее эффективный способ передачи знаний участникам разработки и между ними — личное общение;
  - работающая программа — лучший показатель прогресса разработки.
- Таким образом, эти методы явно ориентированы на итеративную разработку ПО и на МИНИМАЛЬНУЮ формализацию процесса.
- Методология Crystal имеет модификации, предназначенные для выполнения процессов с различным количеством участников и разной критичностью разрабатываемого ПО (критичность ПО определяется возможными последствиями ошибок, которые могут меняться в диапазоне от незначительных финансовых потерь на исправление ошибки до катастрофических).
  - Приведем краткие описания нескольких не упоминавшихся *гибких* методологий: Crystal Clear и FDD.

# Crystal Clear

**Crystal** — семейство методологий, определяющих необходимую степень формализации процесса разработки в зависимости от количества участников и критичности задач.

Методология **Crystal Clear** уступает XP по производительности, зато максимально проста в использовании. Она требует минимальных усилий для внедрения, поскольку ориентирована на человеческие привычки. Считается, что эта методология описывает тот естественный порядок разработки ПО, который устанавливается в достаточно квалифицированных коллективах, если в них не занимаются целенаправленным внедрением другой методологии.

## ● Основные характеристики Crystal Clear:

- итеративная инкрементная разработка;
- автоматическое регрессионное тестирование;
- пользователи привлекаются к активному участию в проекте;
- состав документации определяется участниками проекта;
- как правило, используются средства контроля версий кода.

Помимо Crystal Clear, в семейство Crystal входит еще несколько методологий, предназначенных для выполнения более крупных или более критических проектов. Они отличаются несколько более жесткими требованиями к объему документации и вспомогательным процедурам, таким как управление изменениями и версиями.

# Feature Driven Development

Функционально-ориентированная разработка (Feature Driven Development, FDD) оперирует понятием функции или свойства (feature) системы, достаточно близким к понятию сценария использования, применяемому в RUP. Едва ли не самое существенное отличие — это дополнительное ограничение: **«каждая функция должна допускать реализацию не более чем за две недели»**. То есть если сценарий использования достаточно мал, его можно считать функцией, а если велик, то его надо разбить на несколько относительно независимых функций.

FDD включает пять процессов, причем последние два повторяются для каждой функции:

- разработка общей модели;
- составление списка необходимых функций системы;
- планирование работы над каждой функцией;
- проектирование функции;
- конструирование функции.

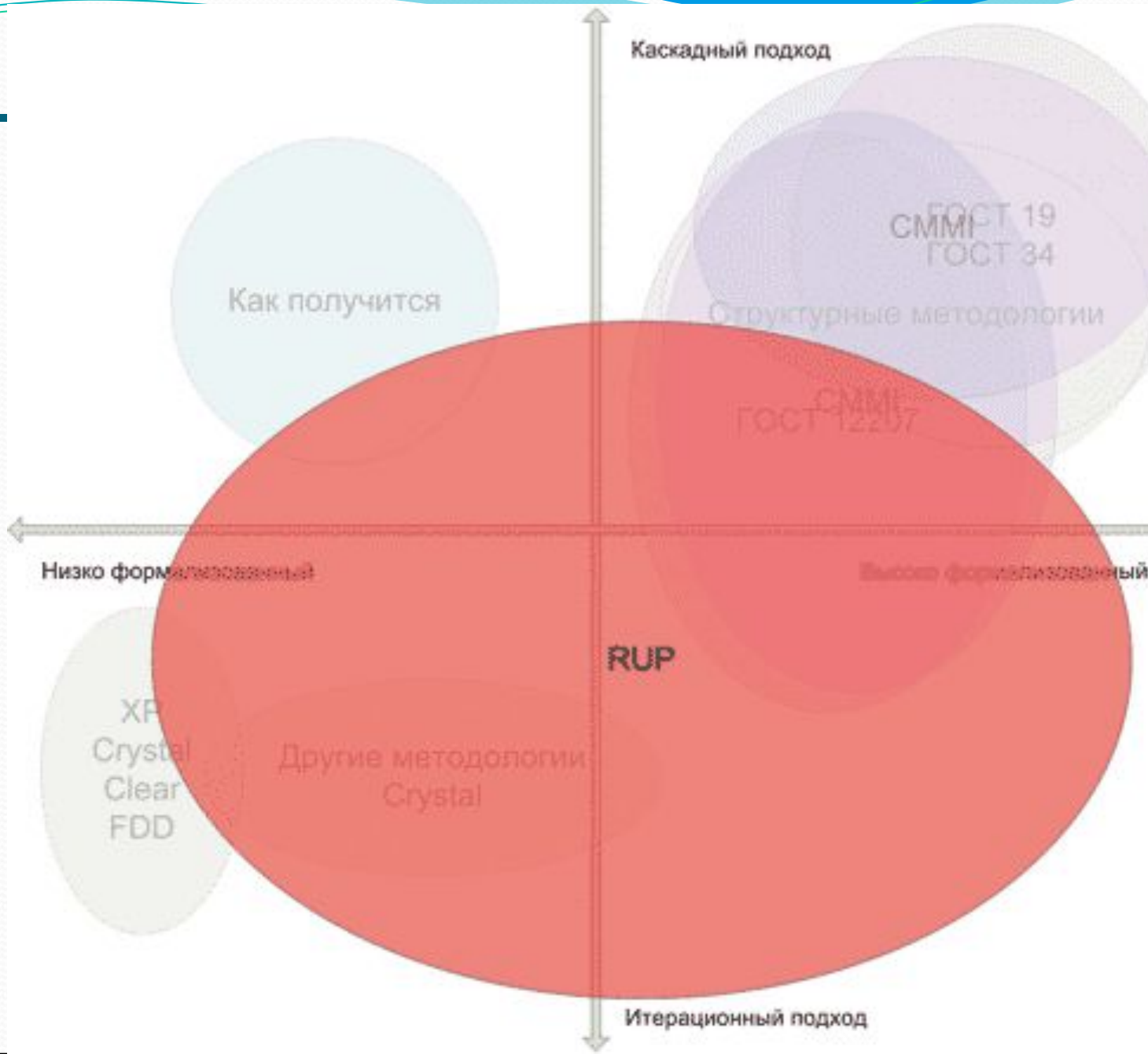
Работа над проектом предполагает частые сборки и делится на итерации, каждая из которых реализуется с помощью определенного набора функций.

Разработчики в FDD делятся на «хозяев классов» и «главных программистов». Главные программисты привлекают хозяев задействованных классов к работе над очередным свойством. Для сравнения: в XP нет персонально ответственных за классы или методы.

# Общие черты гибких методологий

- Список гибких методологий в настоящее время достаточно широк. Методологии XP и Scrum дают весьма полное представление обо всем семействе.
- Практически все гибкие методологии используют итеративный подход, при котором детально планируется только ограниченный объем работ, связанный с выпуском очередного релиза.
- Практически все гибкие методологии ориентированы на максимально неформальный подход к разработке. Если проблему можно решить в ходе обычной беседы, то лучше именно так и поступить. Причем оформлять принятое решение в виде бумажного или электронного документа нужно только тогда, когда без этого невозможно обойтись.

# Методология RUP



RUP — это итеративная методология. Хотя формально обязательность выполнения всех фаз или какого-то минимального числа итераций нигде в RUP не обозначена, весь подход ориентирован на то, что их достаточно много. Ограниченное количество итераций не позволяет в полной мере использовать все преимущества RUP. Вместе с тем RUP можно применять и в практически каскадных проектах, включающих реально всего пару итераций: одну в фазе «Построение», а другую в фазе «Передача». Кстати говоря, в каскадных проектах реально используется именно такое количество итераций.

Ведь проведение испытаний и опытной эксплуатации системы предполагает внесение исправлений, которые могут подразумевать определенные действия, связанные с анализом, проектированием и разработкой, то есть фактически являются еще одним проходом через все фазы разработки.


RUP позволяет разрабатывать только те артефакты и выполнять только те работы и задачи, которые необходимы в конкретном проекте. А выбранные артефакты могут выполняться и рецензироваться с произвольной степенью формальности. Можно требовать детальной проработки и тщательного оформления каждого документа, предоставления столь же тщательно выполненной и оформленной рецензии и даже, следуя старой практике, утверждать каждую такую рецензию на научно-техническом совете предприятия. А можно ограничиться электронным письмом или наброском на бумаге. Кроме того, всегда остается еще одна возможность: сформировать документ в голове, т.е. обдумать соответствующий вопрос и принять конструктивное решение. И если это решение касается только вас, то ограничиться, например, комментарием в коде программы.

Т.е., что касается формальности методологии, то здесь RUP представляет пользователю весьма широкий диапазон возможностей. Если выполнять все работы и задачи, создавать все артефакты и достаточно формально (с официальным рецензентом, с подготовкой полной рецензии в виде электронного или бумажного документа и т.д.) проводить все рецензирования, RUP может оказаться крайне формальной, тяжеловесной методологией.

С другой стороны, RUP - итеративная методология с очень широким диапазоном возможных решений в части формализации процесса разработки.

Итак, RUP, в отличие от большинства других методологий, позволяет в широком диапазоне выбирать степень формализации и итеративности процесса разработки в зависимости от особенностей проектов и разрабатывающей организации.

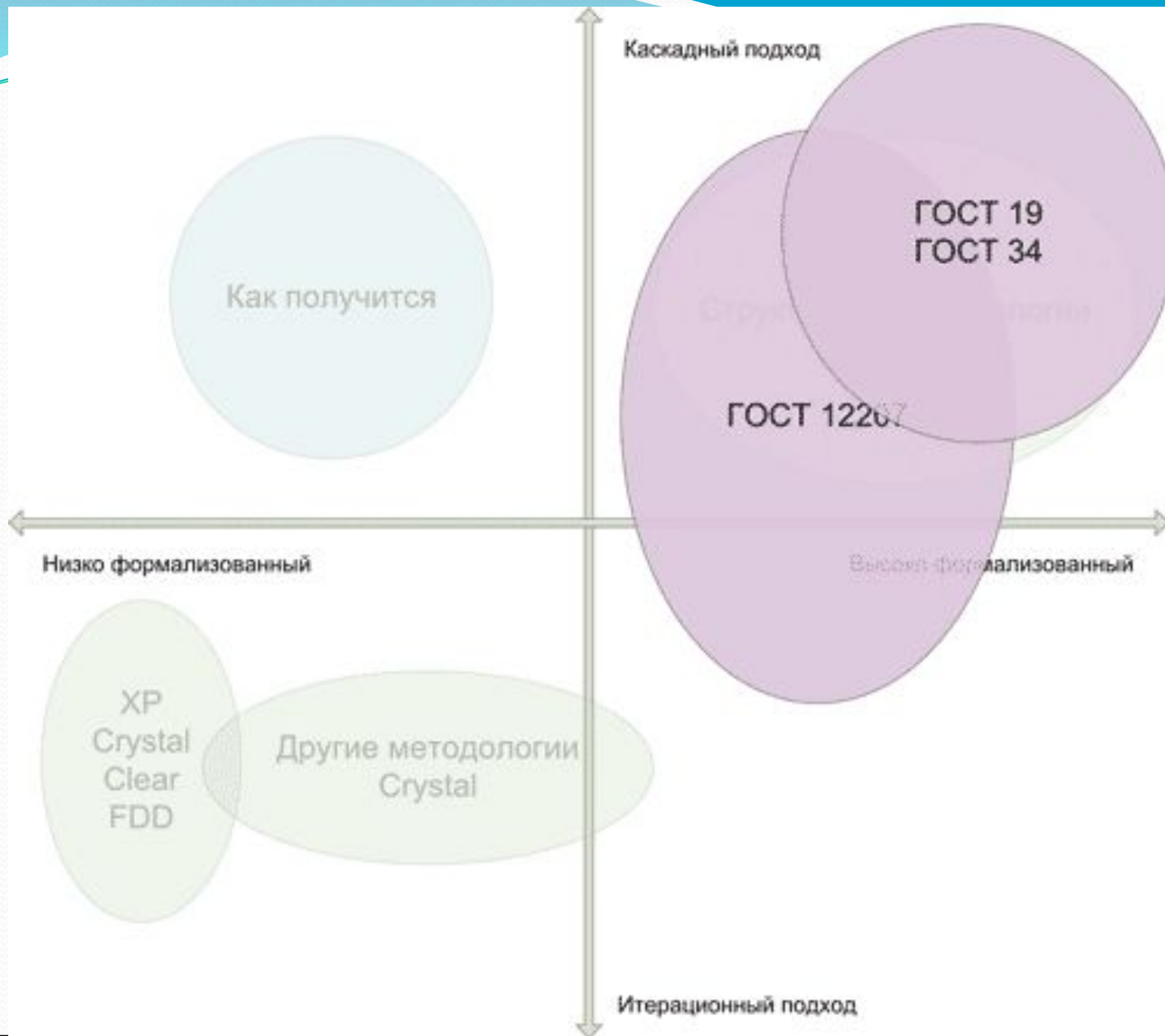




Все эти особенности методов структурного системного анализа ограничили возможности широкого применения соответствующих нотаций и послужили основой для разработки унифицированного языка моделирования UML.

# ГОСТы

- ГОСТы не описывают сами процессы разработки ПО, а только формулируют определенные требования к процессам, которым в той или иной степени соответствуют различные методологии.
- Сравнение требований по тем же критериям, по которым мы сравниваем методологии, поможет сразу определиться с тем, какими методологиями стоит пользоваться, если разработку нужно выполнить в соответствии с ГОСТ.
- В настоящее время в странах СНГ действуют старые ГОСТы 19-й и 34-й серий и более новый ГОСТ Р ИСО МЭК 12207.



# ГОСТы 19-й и 34-й серий

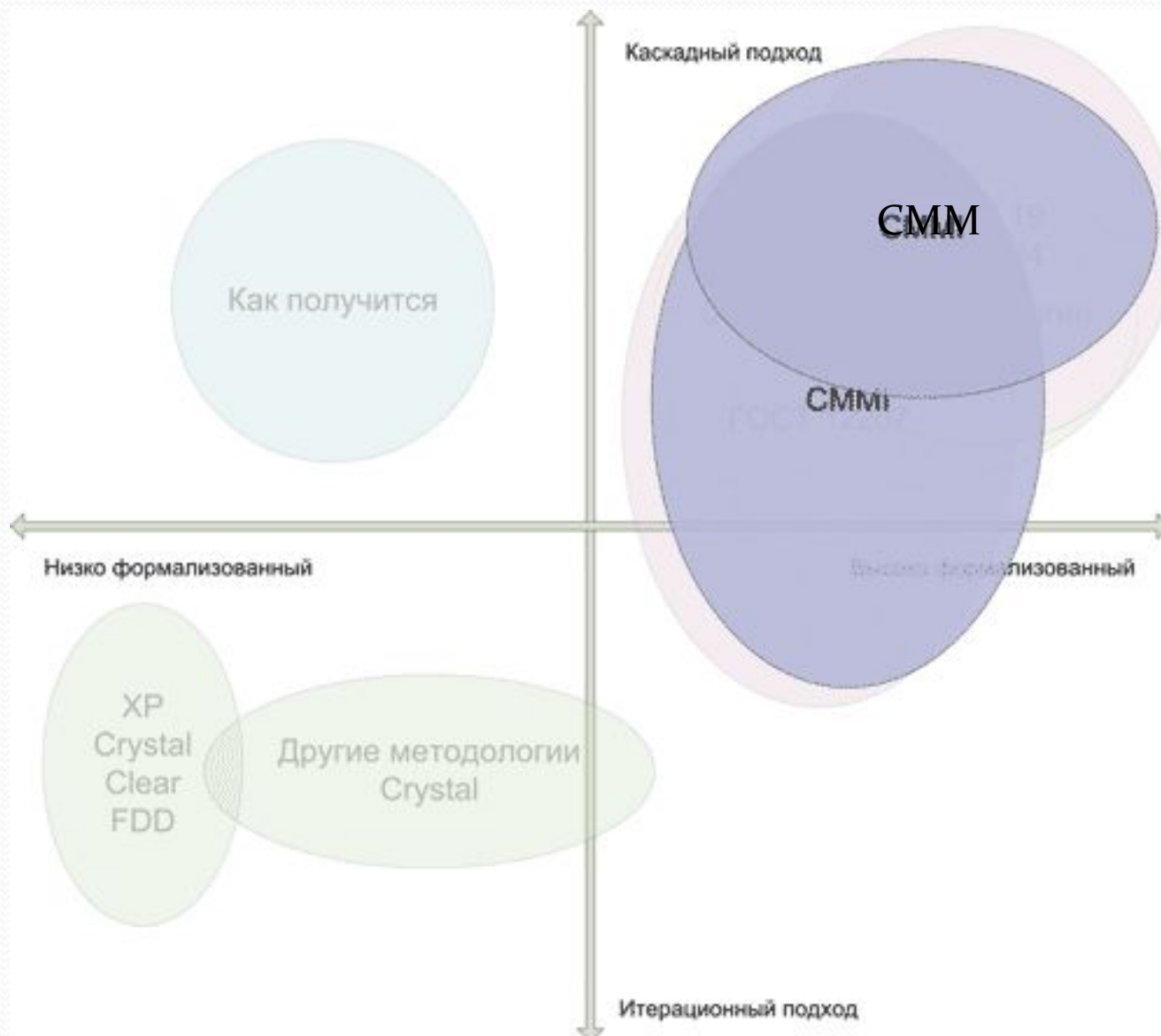
- Жестко ориентированы на каскадный подход к разработке ПО. Разработка в соответствии с этими ГОСТами проводится по этапам, каждый из которых предполагает выполнение строго определенных работ, и завершается выпуском достаточно большого числа весьма формализованных и обширных документов. Т. е. строгое следование этим стандартам не только приводит к каскадному подходу, но и обеспечивает очень высокую степень формализованности разработки.

# ГОСТ 12207

- ГОСТ 12207, в отличие от стандартов 19-й и 34-й серий, описывает разработку ПО как набор основных и вспомогательных процессов, которые могут действовать от начала и до завершения проекта. Модель жизненного цикла может выбираться исходя из особенностей проекта. Таким образом, этот ГОСТ явно не запрещает применение итеративного подхода, но и явно не рекомендует его использование. ГОСТ 12207 также более гибок в части требований к формальности процесса разработки. В нем содержатся только указания на необходимость документирования основных результатов всех процессов, но нет перечней требуемых документов и указаний относительно их содержания.
- Таким образом, ГОСТ 12207 допускает итеративную и менее формализованную разработку ПО.

# Модели зрелости процесса разработки (СММ, СММІ)

- Помимо государственных и международных стандартов, существует несколько подходов к **сертификации процесса разработки**. Наиболее известными из них в СНГ являются СММ и СММІ.



# CMM

## (Capability Maturity Model)

- CMM (Capability Maturity Model) — модель зрелости процессов создания ПО, которая предназначена для оценки **уровня зрелости процесса разработки** в конкретной компании. В соответствии с этой моделью имеется пять уровней зрелости процесса разработки.



# СММ

- Первый уровень соответствует разработке «как получится», когда на каждый проект разработчики идут как на подвиг.
- Второй соответствует более-менее налаженным процессам, когда можно с достаточной уверенностью рассчитывать на положительный исход проекта.
- Третий соответствует наличию разработанных и хорошо описанных процессов, используемых при разработке
- Четвертый — активному использованию метрик в процессе управления для постановки целей и контроля их достижения.
- Пятый уровень означает способность компании оптимизировать процесс по мере необходимости.

# Самостоятельное изучение СМММ

Дополнительный материал

# CMMI

## (Capability Maturity Model Integration)

- В отличие от классической модели СММ, которая была жестко иерархической и допускала только последовательное улучшение процессов по уровням, модель СММІ имеет два измерения – последовательное, такое же как и в СММ, и непрерывное, допускающее совершенствование процессов в организации до некоторой степени в произвольном порядке.

# СММІ

- Ступенчатое представление СММІ позволяет классифицировать организации по пяти уровням зрелости.

# Уровни зрелости процессов по СМММ

- **Начальный уровень** (уровень зрелости 1) – это уровень, на котором, по определению, находится любая компания. На этом уровне разработка ПО ведется более-менее хаотично.
- **Управляемый уровень** (уровень зрелости 2) – здесь уже появляются политики и процедуры организации процессов, утвержденные на уровне компании. Но в полной мере процессы существуют лишь в рамках отдельных проектов.
- **Определенный уровень** (уровень зрелости 3) – здесь появляется стандартный процесс на уровне всей компании в целом. Это большой и постоянно пополняющийся набор активов процесса – шаблонов документов, моделей жизненного цикла, программных средств, практик и пр. Любой конкретный процесс получается вырезкой, из этого стандартного.
- **Управляемый количественно уровень** (уровень зрелости 4) подразумевает появление системы измерений в компании, которые происходят на базе стандартного процесса и позволяют количественно управлять разработкой.
- **Оптимизирующийся уровень** (уровень зрелости 5) подразумевает постоянное улучшение процессов разработки, как постепенных, пошаговых, так и революционных. При этом данные изменения оказываются не вынужденными, а упреждающими проблемы и трудности. Процесс совершенствуется сам и постоянно – есть, реализованы соответствующие механизмы.

# Области усовершенствования.

## Уровень зрелости 2

- Управление требованиями
- Планирование проекта
- Наблюдение за проектом и контроль
- Управление договоренностями с поставщиком
- Измерения и анализ
- Проверка процессов и продуктов на соответствие стандартам
- Конфигурационное управление

# Области усовершенствования.

## Уровень зрелости 3

- Разработка требований
- Техническое решение
- Сборка и поставка продукта
- Проверка продукта на соответствие требованиям (верификация)
- Проверка продукта на соответствие предназначению (валидация)
- Фокусирование на процессах организации
- Определение процессов организации
- Организация обучения
- Комплексное управление проектом
- Управление рисками
- Управление объединенной командой
- Комплексное управление работой с поставщиком
- Принятие решений: оценка альтернатив
- Создание в организации условий для совместной работы

# Области усовершенствования.

## Уровень зрелости 4

- Установление показателей выполнения процессов организации
- Управление проектами на основе количественных показателей



# Области усовершенствования.

## Уровень зрелости 5

- Отбор и внедрение улучшений в организацию
- Анализ причин возникновения проблем и предотвращение их появления в будущем

# Целесообразность использования СММІ определяется условиями

- В относительно больших организациях, которые могут позволить себе значительные накладные расходы на внедрение и использование процесса.
- При высокой текучести кадров, когда тем не менее необходимо поддерживать скорость разработки и качество продуктов на достойном уровне.
- В случаях, когда организация выполняет большое количество более или менее однотипных проектов, СММІ позволяет достаточно точно планировать сроки и бюджет, и выполнять проекты в этих рамках.
- Важная, а часто и определяющая причина для внедрения СММІ – возможность официальной сертификации на достижение определенного уровня зрелости. Нередко наличие или отсутствие сертификации по СММІ является решающим фактором в выборе компании-подрядчика.

# СММ и СММІ

- Основной моделью СММ и СММІ является формализация процесса разработки. Они нацеливают разработчиков на внедрение детально описанного в регламентах и инструкциях процесса, который, в свою очередь, не может не требовать разработки большого объема проектной документации для соответствующего контроля и отчетности.
- Связь СММ и СММІ с итеративной разработкой более опосредованная. Формально ни та ни другая не выдвигают конкретных требований к тому, чтобы придерживаться каскадного или итеративного подхода. Однако, по мнению ряда специалистов, СММ в большей степени совместима с каскадным подходом, в то время как СММІ допускает также и применение итеративного подхода.

# СММІ

После появления СММ стали разрабатываться специализированные модели зрелости для создания ИС, для процесса выбора поставщиков и некоторые другие. На их основе была разработана интегрированная модель СММІ (Capability Maturity Model Integration). Кроме того, в СММІ была предпринята попытка преодолеть проявившиеся к тому времени недостатки СММ — преувеличение роли формальных описаний процессов, когда наличие определенной документации оценивалось значительно выше, чем просто хорошо налаженный, но не описанный процесс. Тем не менее СММІ также ориентирован на использование весьма формализованного процесса.

# СММІ

- Модель Capability Maturity Model Integration (СММІ) была разработана в течение 1990-х годов в университете Карнеги-Меллона (Carnegie Mellon University.) совместно с Software Engineering Institute (SEI) и другими организациями. Одним из главных спонсоров разработки стало Министерство обороны США. СММІ была создана путем объединения (отсюда слово Integration в названии) трех более ранних специализированных моделей разработки: CMM for Software (SW-CMM), Electronic Industries Alliance Interim Standard (EIA/IS) 731 и Integrated Product Development Capability Maturity Model (IPD-CMM). Последняя версия спецификации СММІ 1.2 была опубликована в августе 2006 года.
- Цель внедрения СММІ – построить инфраструктуру процессов, устанавливающую общий стандарт выполнения проектов внутри организации. Для каждого отдельного проекта стандартный процесс должен быть модифицирован в соответствии со спецификой этого проекта. При помощи формальных метрик измеряется эффективность процессов, а сами процессы постоянно оптимизируются.

# СММІ

- СММІ не описывает какой-то конкретный процесс разработки. СММІ-совместимым может быть проект с водопадным, итеративным или другим жизненным циклом. Правильнее думать о СММІ как о наборе элементов процессов, приемов и методик, из которых, как из конструктора, нужно собрать законченный процесс.
- Внедрение СММІ в организации может происходить на непрерывной (continuous) или ступенчатой (staged) основе. Содержание модели СММІ в обоих случаях одно и то же, меняется только ее представление. Непрерывное представление дает возможность производить улучшения в отдельных процессных областях в произвольном порядке. Ступенчатое представление определяет четкую последовательность шагов по внедрению СММІ; каждый шаг соответствует достижению так называемого уровня зрелости (maturity level). Организации необходимо принять решение, до какого уровня зрелости она намерена дойти, а также необходимо ли проходить официальную сертификацию на соответствие этому уровню.
- Остановимся подробнее на ступенчатом представлении, поскольку именно оно чаще всего используется на практике.

# СММІ

- Основными элементами модели СММІ являются процессные области, общие и специальные задачи, общие и специальные практики.
- СММІ определяет 22 процессные области, такие как планирование проекта (Project Planning), управление рисками (Risk Management), разработка требований (Requirements Development) и т.д. В ступенчатом представлении процессные области сгруппированы по пяти уровням зрелости (от 1 до 5). В непрерывном представлении каждая процессная область находится на одном из шести (от 0 до 5) уровней производительности (capability level).
- Процессы в каждой процессной области должны достигать ряда целей. Общие цели (generic goals) относятся к нескольким процессным областям. Специальные цели (specific goals) уникальны для своей процессной области.
- Для достижения специальных и общих целей служат специальные и общие практики (practices). Практики – это деятельность или задачи, которые должны быть выполнены для достижения соответствующей цели.

# Структура СММІ (ступенчатое представление)





## СМММ. Процессная область «Планирование проекта» (Project Planning).

В нее входит:

- определение проектных артефактов,
- разбиение работ на отдельные задачи и их оценка,
- планирование необходимых ресурсов,
- составление графика проекта,
- анализ рисков,
- и т.д.
- В результате планирования проекта создается план проекта (project plan) – основной документ для организации и контроля проектных работ, управления бюджетом и оценки доходности, управления изменениями и рисками.

Замечание. План проекта – более широкое понятие, чем график проекта; как правило, он включает в себя график.

## СМММ. Процессная область «Планирование проекта» должна достигать трех специальных целей

- Установить оценки: подготовить реалистичные оценки, такие как трудоемкость и стоимость проекта, для дальнейшего планирования.
- **Разработать проектный план** (*уточним далее*): создать документ, одобренный заинтересованными сторонами, описывающий жизненный цикл проекта, бюджет, ресурсы, риски, график, стратегию обеспечения качества и т.п.
- Получить обязательства участников проекта: участники проекта, ответственные за его выполнение, должны считать проектный план реалистичным и выполнимым, и подтвердить обязательство выполнить свои задачи в рамках заданного графика, ресурсов и качества.

**СММІ. Процессная область «Планирование проекта» должна достигать двух общих целей.**

- Учредить управляемый процесс: общее требование к процессам на уровне зрелости 2.
- Учредить определенный (defined) процесс: общее требование к процессам на уровне зрелости 3.

## **СМММ. Практики, позволяющие достичь цели «разработать проектный план».**

- Установить бюджет и график проекта.
- Идентифицировать риски.
- Спланировать управление данными (определить источники и форматы данных, необходимые процедуры для миграции, репликации и получения данных, требования по безопасности и т.д.).
- Определить необходимые ресурсы.
- Определить требования к профессиональным навыкам сотрудников, запланировать тренинги.
- Запланировать вовлечение всех