

Управление проектом

- Зачем управлять проектом?

История вопроса

- В основе методов управления проектами лежат методики сетевого планирования, разработанные в конце 50-х гг. в США.
- Методики получили широкое распространение не только в странах с рыночной экономикой, но и в странах планово-директивной направленности экономики.
- Использовались в строительном производстве. Именно с них началось возникновение и распространение методов проектного управления.

Примеры

- В 1956 г. фирма «Дюпон», исследуя возможности более эффективного использования принадлежащей фирме вычислительной машины Univac, объединила свои усилия с группой планирования капитального строительства фирмы «Ремингтон Рэнд». Они попытались использовать ЭВМ для составления планов-графиков крупных комплексов работ по модернизации заводов фирмы «Дюпон». В результате был создан рациональный и простой метод описания проекта с использованием ЭВМ. Первоначально он был назван методом Уолкера-Келли, а позже получил название **Метода Критического Пути** – МКП (или **СРМ** – Critical Path Method).

- Параллельно и независимо в военно-морских силах США был создан **метод анализа и оценки программ PERT** (Program Evaluation and Review Technique).
- Разработан корпорацией «Локхид» и консалтинговой фирмой «Буз, Аллен энд Гамильтон» для реализации проекта разработки ракетной системы «Поларис», объединяющего около 3800 основных подрядчиков и состоящего из 60 тыс. операций.

- Метод PERT позволил руководству программы точно знать, что требуется делать в каждый момент времени и кто именно должен это делать, а также вероятность своевременного завершения отдельных операций.
- Руководство программой оказалось настолько успешным, что проект удалось завершить на два года раньше запланированного срока.
- Благодаря такому успешному началу данный метод управления вскоре стал использоваться для планирования

- Для управления проектом сооружения гидроэлектростанции на реке Черчилль в Ньюфаундленде (полуостров Лабрадор).
- Стоимость проекта составила 950 млн. дол. Гидроэлектростанция строилась с 1967 по 1976 г.
- Включал более 100 строительных контрактов, причем стоимость некоторых из них достигала 76 млн. дол. В 1974 г. ход работ по проекту опережал расписание на 18 месяцев и укладывался в плановую оценку затрат. Заказчиком проекта была корпорация Churchill Falls Labrador Corp., которая для разработки проекта и управления строительством наняла фирму Acres Canadian Betchel.

- По существу, значительный выигрыш по времени образовался от применения **новых методов планирования и управления на основе информационных технологий** и возможностей вычислительной техники. Однако первые ЭВМ были дороги и доступны только крупным организациям.

- Этап наиболее бурного развития систем для управления проектами начался с появлением **персональных компьютеров**, когда компьютер стал рабочим инструментом для широкого круга руководителей.

- Например, с 1993 по 1998 г. в государстве Катар на берегу Персидского залива японской компанией «Chiyoda» при участии примерно 2000 компаний-соисполнителей из более чем 50 стран мира был возведен завод по производству сжиженного газа и морской терминал по его обслуживанию (Qatargas LNG Plant).
- Бюджет порядка \$1,7 млрд.
- Проект осуществлялся с применением профессионального управления и современных информационных технологий, включая телекоммуникации через спутник Intel Sat и глобальной сети Интернет: Electronic Data Management System (EDMS), Global Communication System, Project Material Management System, New Project Management Tools, Project IT.

- На инжиниринговые и управленческие работы было затрачено около 40 % бюджета проекта: завод был спроектирован с помощью трехмерной системы автоматизации проектирования, построен в компьютере на виртуальной строительной площадке, при этом была испытана компьютерная технология и организация строительно-монтажных работ.
- После этого было проведено комплексное испытание виртуального завода, внесены соответствующие коррективы в проект и выданы на печать необходимая проектно-сметная документация и материалы по управлению этим проектом, включая контракты, планы, графики, спецификации, комплектные поставки и т. п.
- Только после того, как вся работа на компьютерах была проделана, приступили к работам на строительной площадке.

- Обращаясь к мировому опыту, следует отметить, что в ходе постепенного развития системы управления проектами (как самостоятельной области профессиональной деятельности) в конечном итоге были созданы собственные унифицированные механизмы, методологии, инструментарий и стандарты.

- Так, например, создана единая **Международная ассоциация управления проектами** – IPMA с центром в г. Цюрих, Швейцария.

- Самое широкое распространение получила **процессная модель**, которая используется в таких наиболее известных документах, излагающих методологические основы управления проектами, как **Project Management Body of Knowledge (PMBOK)** Американского института управления проектами (PMI), многими признаваемый международным стандартом де-факто и
- стандарт ISO 10006:1997, придавший ряду наиболее важных положений PMBOK статус стандарта де-юре. Заменявший первый PMBOK редакции 1987 г. A Guide to the Project Management Body of Knowledge (PMBOK Guide) редакции 1996 года признан национальным стандартом США ANSI/PMI 99-001-2000.

Общепринятые сокращения в мировой системе управления проектами

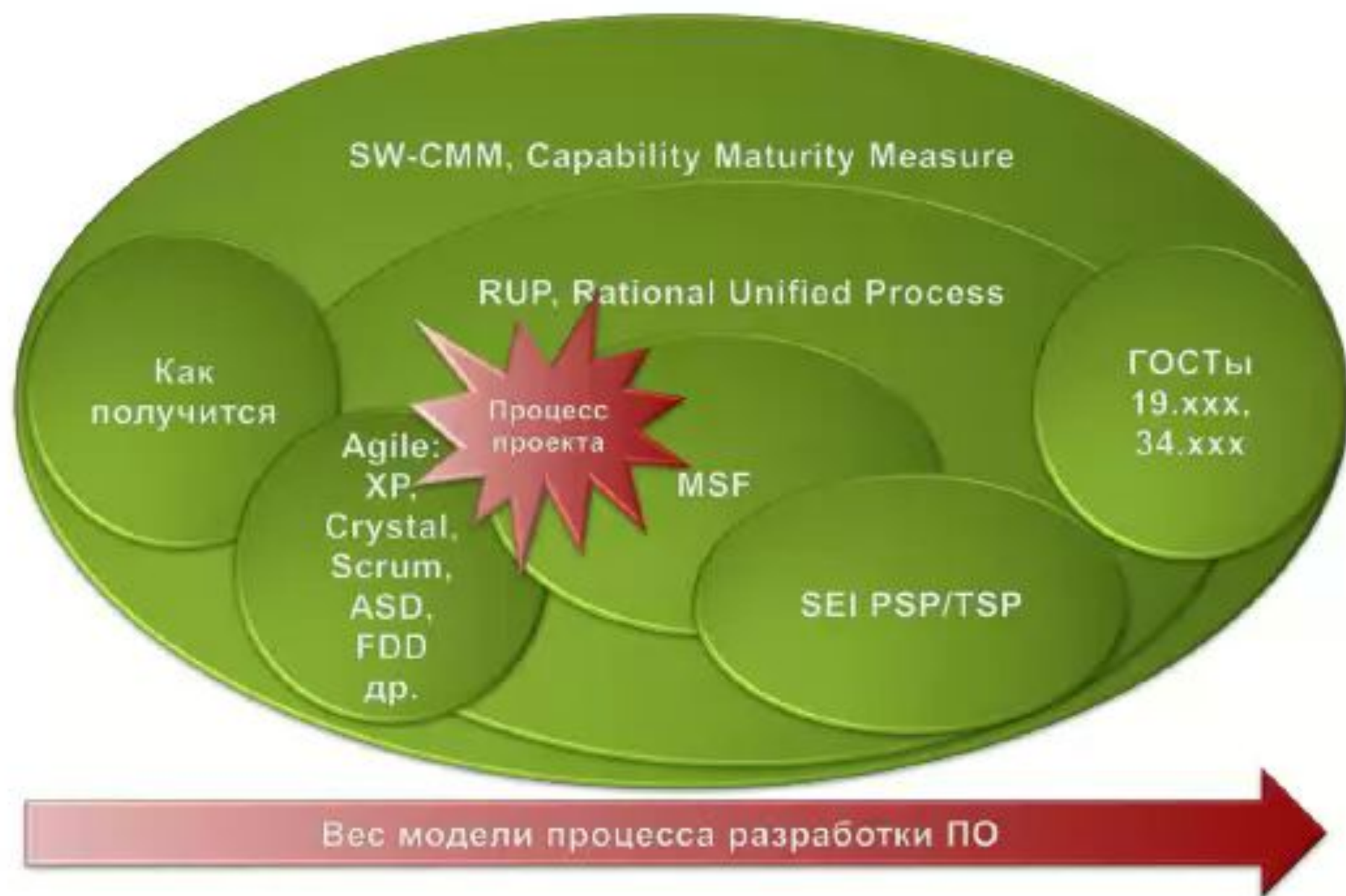
Сокращение	Расшифровка
PM	Project Management
IPMA	International Project Management Association
PMI	Project Management Institute (США)
AIPM	Australian Institute for Project Management (Австралия)
APM	Association for Project Managers (Великобритания)
СОВНЕТ	Ассоциация управления проектами (Россия)
ENAA	Engineering Advancement Association of Japan (Япония)
GPM	Deutsche Gesellschaft für Projektmanagement
ICB IPMA	International Competence Baseline IPMA
NCB	National Competence Baseline
PM BoK	Project Management Body of Knowledge
PMBOK	Project Management Body of Knowledge PMI (США)

- По данным Международной ассоциации управления проектами (IPMA) использование современной методологии и инструментария управления проектами позволяет обычно сэкономить порядка 20–30 % времени и около 15–20 % средств, затрачиваемых на осуществление проектов и программ.

- Существует целый ряд стандартов, регламентирующих ЖЦ ПО, а в некоторых случаях и процессы разработки.
- Значительный вклад в теорию проектирования и разработки информационных систем внесла компания IBM, предложив еще в середине 1970-х годов методологию BSP (Business System Planning - методология организационного планирования).

Методики (методологии) управления ИТ-проектами

- Модели (методики, 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.
- Делятся на:
 - Тяжеловесные: RUP, MSF
 - Легковесные или agile-методики.



ГОСТы

- ГОСТ 19 «Единая система программной документации» и 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.
- Таким образом, строгое следование этим гостам не только приводит к водопадному подходу, но и требует очень высокой степени формализованности разработки.

SW-CMM

- В середине 80-х годов 20-го столетия Министерство обороны США крепко задумалось о том, как выбирать разработчиков ПО при реализации крупномасштабных программных проектов. По заказу военных Институт программной инженерии, входящий в состав Университета Карнеги-Меллона, разработал SW-CMM, Capability Maturity Model for Software в качестве эталонной модели организации разработки программного обеспечения.

Модель определяет 5 уровней зрелости процесса разработки ПО:

- **Начальный** — процесс разработки носит хаотический характер. Определены лишь немногие из процессов, и успех проектов зависит от конкретных исполнителей.
- **Повторяемый** — установлены основные процессы управления проектами: отслеживание затрат, сроков и функциональности. Упорядочены некоторые процессы, необходимые для того, чтобы повторить предыдущие достижения на аналогичных проектах.

- **Определенный** — процессы разработки ПО и управления проектами описаны и внедрены в единую систему процессов компании. Во всех проектах используется стандартный для организации процесс разработки и поддержки программного обеспечения, адаптированный под конкретный проект.
- **Управляемый** — собираются детальные количественные данные по функционированию процессов разработки и качеству конечного продукта. Анализируется значение и динамика этих данных.
- **Оптимизируемый** — постоянное улучшение процессов основывается на количественных данных по процессам и на пробном внедрении новых идей и технологий.

- Документация с полным описанием SW-CMM занимает около 500 страниц и определяет набор из 312 требований, которым должна соответствовать организация, если она планирует аттестоваться по этому стандарту на 5-ый уровень зрелости.

RUP

- Один из самых известных процессов, использующих итеративную модель разработки – **Rational Unified Process (RUP)**.
- Создан во второй половине 1990-х годов в компании Rational Software.
- Основныи разработчики Филипп Крачтен (Philippe Kruchten), Грейди Буч (Grady Booch), Джеймс Рамбо (James Rumbaugh) и Айвар Якобсон (Ivar Jacobson).
- Последние трое являются также создателями нотации UML.

- Rational Unified Process предлагает **итеративную** модель разработки, включающую 4 фазы: начало, исследование, построение и внедрение.
- Каждая фаза может быть разбита на **этапы (итерации)**, в результате которых выпускается версия для внутреннего или внешнего использования.
- Прохождение через фазы - **цикл 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.
- Суть работы в рамках RUP - это создание и сопровождение моделей на базе 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.

Рабочий процесс в RUP

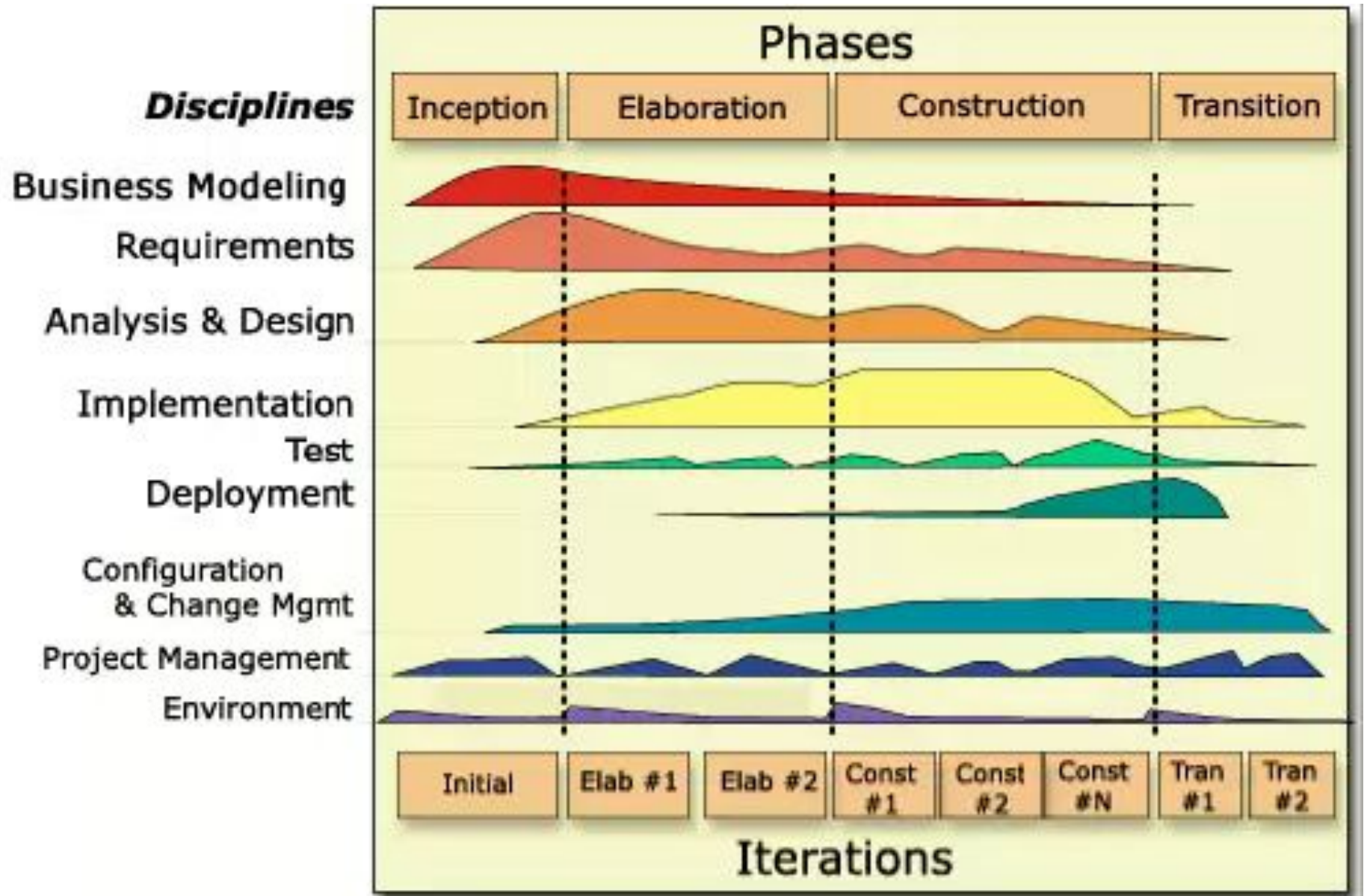
- В терминах RUP участники проектной команды создают так называемые **артефакты** (work products), выполняя **задачи** (tasks) в рамках определенных **ролей** (roles).
- Артефактами являются спецификации, модели, исходный код и т.п.
- Задачи разделяются по девяти процессным областям, называемым **дисциплинами** (discipline).

- В дисциплины входят:
- **Бизнес-моделирование** (Business Modeling) – исследование и описание существующих бизнес-процессов заказчика, а также поиск их возможных улучшений.
- **Управление требованиями** (Requirements Management) – 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.

- **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.

- В ходе жизненного цикла проекта распределение усилий проектной команды между дисциплинами постоянно меняется.
- Например, как правило, в начале проекта большая часть усилий затрачивается на анализ и дизайн, а ближе к завершению – на реализацию и тестирование системы. Однако в общем случае задачи из всех девяти дисциплин выполняются параллельно.
- Рисунок иллюстрирует выполнение проекта среднего размера.



- Для полноценного внедрения RUP организация должна затратить значительные средства на обучение сотрудников.
- При этом попытка обойтись своими силами скорее всего будет обречена на неудачу – необходимо искать специалиста по процессам (process engineer) с соответствующим опытом или привлекать консультантов.

MSF

- Посредством пакета руководств MSF (Microsoft Solutions Framework) гигант IT индустрии решил поделиться опытом и накопленной информацией в области проектирования, разработки, внедрения и сопровождения IT –проектов.
- База знаний MSF состоит из оригинальных моделей, методов и взглядов на такие области знаний как управление проектами, персоналом, планирование, анализ рисков и другие смежные дисциплины.

Структурно пакет руководств MSF разделён на пять документов, так называемых «белых книг», каждый из которых охватывает определенную дисциплину или модель MSF :

- «Модель процессов MSF»,
- «Модель проектной группы MSF»,
- «Дисциплина управления проектами MSF»,
- «Дисциплина управления рисками MSF» и
- «Дисциплина управления подготовкой MSF».

PSP/TSP

- Одна из последних разработок Института программной инженерии Personal Software Process / Team Software Process.
- Personal Software Process определяет требования к компетенциям разработчика.

Согласно этой модели каждый программист должен уметь:

- учитывать время, затраченное на работу над проектом;
- учитывать найденные дефекты;
- классифицировать типы дефектов;
- оценивать размер задачи;
- осуществлять систематический подход к описанию результатов тестирования;
- планировать программные задачи;
- распределять их по времени и составлять график работы.
- выполнять индивидуальную проверку проекта и архитектуры;
- осуществлять индивидуальную проверку кода;
- выполнять регрессионное тестирование.

- Team Software Process делает ставку на самоуправляемые команды численностью 3-20 разработчиков.
- Команды должны:
 - установить собственные цели;
 - составить свой процесс и планы;
 - отслеживать работу;
 - поддерживать мотивацию и максимальную производительность.
- Последовательное применение модели PSP/TSP позволяет сделать нормой в организации пятый уровень CMM.

Agile

- Основная идея всех гибких моделей заключается в том, что применяемый в разработке ПО процесс должен быть **адаптивным**.
- Они декларируют своей высшей ценностью **ориентированность на людей** и их взаимодействие, а не на процессы и средства.
- По сути, так называемые, гибкие методологии это не методологии, а **набор практик**, которые могут позволить (а могут и нет) добиваться эффективной разработки ПО, основываясь на итеративности, инкрементальности, самоуправляемости команды и адаптивности процесса.

- В течение 1990-х годов все больше разработчиков ПО начинали искать альтернативу традиционным, как правило, основанным на модели водопада, процессам разработки. К 2000 году существовало уже целое множество так называемых **легковесных (lightweight) методологий.**

- В 2001 году группа создателей и экспертов по различным легковесным методологиям провела семинар, на котором были сформулированы основные принципы гибкой разработки ПО (так называемый Agile Manifesto) – манифест гибкой разработки.
- На том же семинаре было предложено новое название легковесных методологий – гибкая разработка (agile software development).

Общими особенностями гибких методологий являются:

- **Ориентированность на людей** – как разработчиков, так и заказчиков. Считается, что умение собрать в проектной команде «правильных» людей определяет успех или неудачу проекта в значительно большей степени, чем любые процессы или технологии.
- Использование **устных** обсуждений вместо **формальных** спецификаций везде, где это возможно. Обсуждения должны быть главным способом коммуникации как с

- **Итеративная** разработка с возможно более короткой (в разумных пределах) продолжительностью итерации, при этом в результате каждой итерации выпускается полноценная работающая версия продукта.
- **Ожидание изменений** – в гибком процессе проектная команда не пытается зафиксировать требования в начале проекта и затем следовать жестко определенному плану. Изменения могут быть сделаны на сколь угодно позднем этапе проекта.

Принципы, которые разъясняет Agile Manifesto:

- удовлетворение клиента за счёт ранней и бесперебойной поставки ценного программного обеспечения;
- приветствие изменений требований даже в конце разработки (это может повысить конкурентоспособность полученного продукта);
- частая поставка рабочего программного обеспечения (каждый месяц или неделю или ещё чаще);
- тесное, ежедневное общение заказчика с

- проектом занимаются мотивированные личности, которые обеспечены нужными условиями работы, поддержкой и доверием;
- рекомендуемый метод передачи информации — личный разговор (лицом к лицу);
- работающее программное обеспечение — лучший измеритель прогресса;
- спонсоры, разработчики и пользователи должны иметь возможность поддерживать постоянный темп на неопределённый срок;

- постоянное внимание улучшению технического мастерства и удобному дизайну;
- простота — искусство не делать лишней работы;
- лучшие технические требования, дизайн и архитектура получаются у самоорганизованной команды;
- постоянная адаптация к изменяющимся обстоятельствам.

eXtreme Programming или XP (экстремальное программирование)

- Методология XP была создана Кентом Беком (Kent Beck) в 1996 году в ходе попытки спасти провальный проект по разработке системы расчета зарплаты для компании Крайслер.
- В 2000 году проект был закрыт, но XP к тому времени уже получила известность и начала распространяться среди разработчиков ПО.
- XP наследует все общие принципы гибких методологий, достигая их при помощи двенадцати инженерных практик.

- Она описывается как 12 практик: игра в планирование, короткие релизы, метафоры, простой дизайн, переработки кода (refactoring), разработка "тестами вперед", парное программирование, коллективное владение кодом, 40-часовая рабочая неделя, постоянное присутствие заказчика и стандарты кода.
- Интерес к XP рос снизу вверх, от разработчиков и тестировщиков, замученных тягостным процессом, документацией, метриками и прочим формализмом. Они не отрицали дисциплину, но не желали бессмысленного соблюдения формальных требований. И искали новые быстрые и гибкие подходы к разработке высококачественных программ.

Задание:

- Нарисовать дом

Crystal Clear

- Разработчик Алистер Коуберн
- Crystal - семейство методологий, определяющих необходимую степень формализации процесса разработки в зависимости от количества участников и критичности задач.
- Уступает XP по производительности, зато максимально проста в использовании. Требует минимальных усилий для внедрения, так как ориентирована на человеческие привычки.
- Считается, что она описывает тот естественный порядок разработки ПО, который устанавливается в достаточно квалифицированных коллективах, если в них не занимаются целенаправленным внедрением другой методологии.

Основные характеристики методологии:

- 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 (FDD, Feature Driven Development).
- На самом деле используемое в FDD понятие функции или свойства (feature) системы достаточно близко к понятию прецедента использования, используемому 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. Если же велик, то его надо разбить на несколько относительно независимых функций.

FDD включает 5 процессов, последние два из которых повторяются для каждой функции:

1. Разработка общей модели.
2. Составление списка необходимых функций системы.
3. Планирование работы над каждой функцией.
4. Проектирование функции.
5. Конструирование функции.

- Разработчики в FDD делятся на "хозяев классов" – class owners и "главных программистов" chief programmers.
- Главные программисты привлекают хозяев задействованных классов к работе над очередным свойством. Для сравнения, в XP нет персонально ответственных за классы или методы. К работе над любым классом может привлекаться любой из участников разработки.
- Работа над проектом предполагает частые сборки и делится на итерации, каждая из которых предполагает реализацию

OpenUP

- Итеративно-инкрементальный метод разработки ПО. Позиционируется как легкий и гибкий вариант RUP.
- Базовый процесс OpenUP является независимым от инструментов, малорегламентированным процессом, который может быть расширен для удовлетворения потребностей широкого диапазона типов проектов.

В основу OpenUP положены следующие основные **принципы**:

- Совместная работа с целью согласования интересов и достижения общего понимания;
- Развитие с целью непрерывного обеспечения обратной связи и совершенствования проекта;
- Концентрация на архитектурных вопросах на ранних стадиях для минимизации рисков и организации разработки;
- Выравнивание конкурентных преимуществ для максимизации потребительской ценности для заинтересованных лиц.

Scrum

- Scrum — методология управления разработкой информационных систем для гибкой разработки программного обеспечения.
- Scrum чётко делает акцент на качественном контроле процесса разработки.
- Кроме управления проектами по разработке ПО Scrum может также использоваться в работе команд поддержки программного обеспечения (software support teams), или как подход управления разработкой и сопровождением программ: Scrum of Scrums.

- Это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые **спринтами** (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет.
- Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всём его протяжении. При этом строго фиксированная небольшая длительность спринта придаёт процессу разработки предсказуемость и гибкость.

- **Спринт** — итерация, в ходе которой создаётся функциональный рост программного обеспечения.
- Жёстко фиксирован по времени. Длительность одного спринта от 2 до 4 недель. В отдельных случаях, к примеру согласно Scrum стандарту Nokia, длительность спринта должна быть не более 6 недель.
- Считается, что чем короче спринт, тем более гибким является процесс разработки, релизы выходят чаще, быстрее поступают отзывы от потребителя, меньше времени тратится на работу в неправильном направлении

- По методике Scrum в производственном процессе есть определенные роли, разбитые на 2 группы «свиней» и «кур».
- Названия были использованы из-за шутки:
- Свинья идёт по дороге. Курица смотрит на нее и говорит: «А давай откроем ресторан!» Свинья смотрит на курицу и отвечает: «Хорошая идея, и как ты хочешь его назвать?» Курица думает и говорит: «Почему бы не назвать 'Яичница с беконом'?». «Так не пойдёт», — отвечает свинья, «ведь тогда мне придётся полностью посвятить себя проекту, а ты будешь вовлечена только частично.»

Основные роли:

- Скрам-мастер (ScrumMaster) – проводит совещания следит за соблюдением всех принципов, разрешает противоречия и защищает команду от отвлекающих факторов. Только ведет скрам-процесс.
- Владелец проекта (Product Owner) – представляет интересы конечных пользователей и других заинтересованных в продукте сторон.
- Скрам-команда (Scrum Team) – кросс-функциональная команда разработчиков проекта, состоящая из специалистов разных профилей: тестировщиков, архитекторов, аналитиков, программистов и т. д. Размер команды в идеале составляет 7 ± 2 человека. Команда является единственным полностью вовлечённым участником разработки и отвечает за результат.

Неосновные роли:

- Пользователи (Users)
- Клиенты, Продавцы (Stakeholders) – лица, которые инициируют проект и для кого проект будет приносить выгоду. Они вовлечены в скрам только во время обзорного совещания по спринту (Sprint Review).
- Управляющие (Managers) – люди, которые управляют персоналом.
- Эксперты-консультанты (Consulting Experts)

Ежедневное совещание (Daily Scrum meeting):

- начинается точно вовремя;
- все могут наблюдать, но только основные говорят;
- длится не более 15 минут;
- проводится в одном и том же месте в течение спринта.

В течение совещания каждый член команды отвечает на 3 вопроса:

1. Что сделано с момента предыдущего ежедневного совещания?
2. Что будет сделано с момента текущего совещания до следующего?
3. Какие проблемы мешают достижению целей спринта? (Над решением этих проблем работает скрам мастер. Обычно это решение проходит за рамками ежедневного совещания и в составе лиц, непосредственно затронутых данным препятствием.)

Выбор методологии

Вес модели	Плюсы	Минусы
Тяжелые	Процессы рассчитаны на среднюю квалификацию исполнителей. Большая специализация исполнителей. Ниже требования к стабильности команды. Отсутствуют ограничения по объему и сложности выполняемых проектов.	Требуют существенной управленческой надстройки Более длительные стадии анализа и проектирования. Более формализованные /коммуникации. 8
Легкие	Меньше непроизводительных расходов, связанных с управлением проектом, рисками, изменениями, конфигурациями. Упрощенные стадии анализа и проектирования, основной упор на разработку функциональности, совмещение ролей. Неформальные коммуникации.	Эффективность сильно зависит от индивидуальных способностей, требуют более квалифицированной, универсальной и стабильной команды. Объем и сложность выполняемых проектов

- Алистер Коуберн, один из авторов «Манифеста гибкой разработки ПО» проанализировал очень разные программные проекты, которые выполнялись по разным моделям от совершенно облегченных и «гибких» до тяжелых (СММ-5) за последние 20 лет.
- Он не обнаружил корреляции между успехом или провалом проектов и моделями процесса разработки, которые применялись в проектах.
- Сделал вывод о том, что эффективность разработки ПО **не зависит от модели процесса**, а также о том, что :
 - У каждого проекта должна быть своя модель процесса разработки.
 - У каждой модели — свое время.

- Это означает, что **не существует единственного правильного процесса разработки ПО**, в каждом новом проекте процесс должен определяться каждый раз заново, в зависимости от проекта, продукта и персонала, в соответствие с «Законом 4-х П»

