
Лекция

**УПРАВЛЕНИЕ ПРОЕКТАМИ.
ОПРЕДЕЛЕНИЯ И КОНЦЕПЦИИ**

Король Иван Андреевич

-
1. Проект – основа инноваций
 2. Критерии успешности проекта
 3. Проект и организационная структура компании
 4. Организация проектной команды
 5. Жизненный цикл проекта. Фазы и продукты
 6. Выводы
 7. Контрольные вопросы
-

■ Классическое управление проектами выделяет два вида организации человеческой деятельности: **операционная** и **проектная**.

■ **Операционная деятельность** применяется, когда внешние условия хорошо известны и стабильны, когда производственные операции хорошо изучены и неоднократно испытаны, а функции исполнителей определены и постоянны.

■ В этом случае основой эффективности служат узкая специализация и повышение компетенции.

■ «Если водитель трамвая начнет искать новые пути, жди беды».

-
- Там,
 - а) где разрабатывается новый продукт, внешние условия и требования к которому постоянно меняются,
 - б) где применяемые производственные технологии используются впервые,
 - в) где постоянно требуются поиск новых возможностей,
 - г) интеллектуальные усилия и творчество,
 - там требуются **ПРОЕКТЫ**.
-

-
-
- **Проект** – временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов.
-

■ У операционной и проектной деятельности есть **ряд общих характеристик:**

- выполняются людьми,
- ограничены доступностью ресурсов,
- планируются,
- исполняются
- и управляются.

■ Операционная деятельность и проекты различаются, главным образом, тем, что **операционная деятельность** – это продолжающийся во времени и повторяющийся процесс, в то время как **проекты** являются временными и уникальными.

-
- **Ограничение по срокам** означает, что у любого проекта есть четкое начало и четкое завершение. ***Завершение наступает:***
 - когда достигнуты цели проекта;
 - или осознано, что цели проекта не будут или не могут быть достигнуты;
 - или исчезла необходимость в проекте, и он прекращается.
-

-
- **Уникальность** также важное отличие проектной деятельности от операционной.
 - Если бы результаты проекта не носили уникальный характер, работу по их достижению можно было бы четко регламентировать, установить производственные нормативы и реализовывать в рамках операционной деятельности (конвейер).
 - **Задача проекта** – достижение конкретной бизнес-цели.
 - **Задача операционной деятельности** – обеспечение нормального течения бизнеса.
-

■ **Проект** – это средство стратегического развития



-
- **Цель** – описание того, что мы хотим достичь.
 - **Стратегия** – констатация того, каким образом мы собираемся эти цели достигать.
 - **Проекты преобразуют стратегии в действия, а цели в реальность.**
 - Таким образом, каждая работа, которую выполняет конкретный сотрудник, привязывается к достижению **стратегических целей организации.**
-

-
- Проекты объединяются в программы.
 - **Программа** — ряд связанных друг с другом проектов, управление которыми координируется для достижения преимуществ и степени управляемости, недоступных при управлении ими по отдельности.
 - Проекты и программы объединяются в портфели.
 - **Портфель** — набор проектов или программ и других работ, объединенных вместе с целью эффективного управления данными работами для достижения стратегических целей.
-

-
- Проекты и управление ими существовали всегда.
 - 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**.

■ ***Проект это основа инноваций.***

■ Сделать то, до чего другие компании еще не додумались, сделать это как можно быстрее, иначе это сделают другие.

■ Предложить потребителю более качественный продукт или такой продукт, потребность в котором потребитель даже не может пока осознать.

- **Задача проекта** – достижение конкретной бизнес-цели, при соблюдении ограничений «железного треугольника»

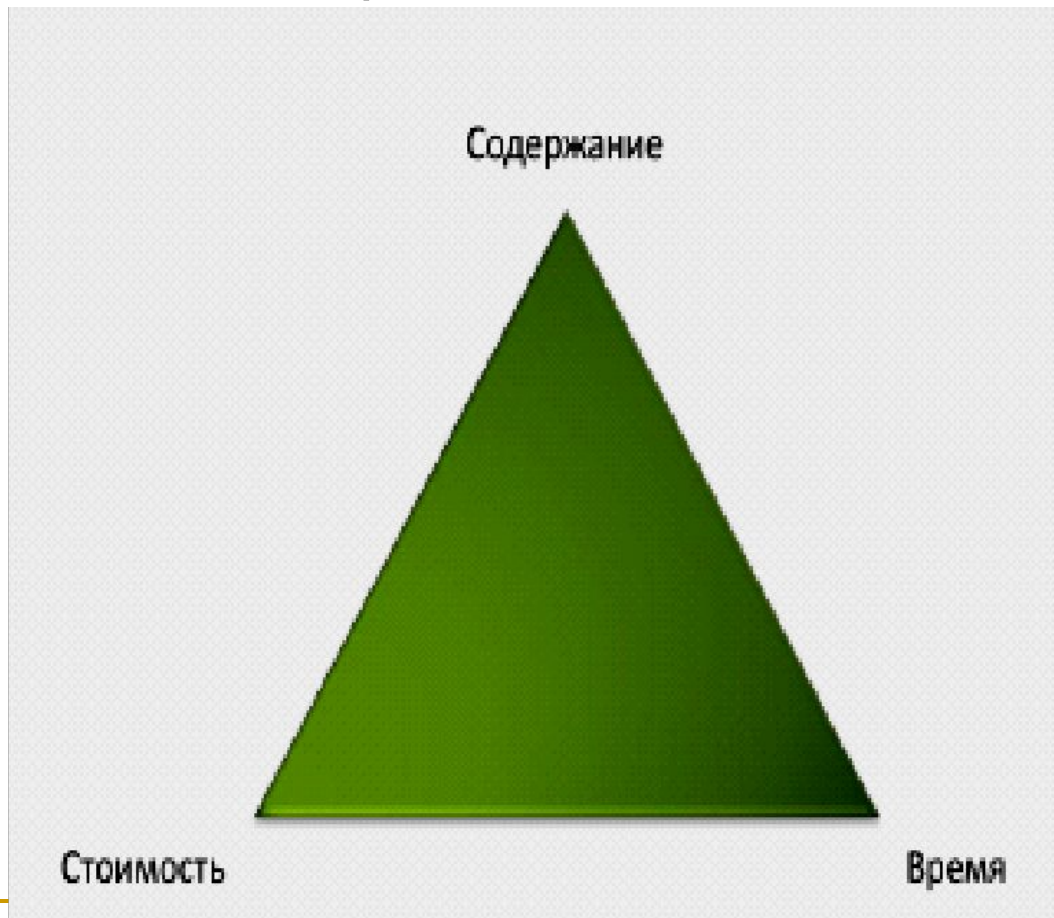


Рис. 5.2. «Железный треугольник» ограничений проекта

-
- Это означает, что ни один из углов треугольника не может быть изменен без оказания влияния на другие.
 - **Например**, чтобы уменьшить время, потребуется увеличить стоимость и/или сократить содержание.
 - Согласно текущей редакции стандарта РМВОК [1], проект считается успешным, если **удовлетворены все требования заказчика и участников проекта**.
-

-
- Поэтому у проекта разработки ПО сегодня не три, а четыре **фактора успеха**:
 - Выполнен в соответствие со спецификациями.
 - Выполнен в срок.
 - Выполнен в пределах бюджета.
 - *Каждый участник команды уходил с работы в 18:00 с чувством успеха.*
 - Этот четвертый фактор успеха должен стать воспроизводимым, если **предприятие хочет быть эффективным.**
-

-
- Для успешного проекта характерно постоянное ощущение его участниками чувства **удовлетворения и гордости** за результаты своей работы, чувства оптимизма.
 - ***Нет ничего более губельного для проекта, чем равнодушие или уныние его участников.***
 - **Эффективность** это отношение полученного результата к произведенным затратам.
 - Нельзя рассматривать эффективность, исходя только из результативности: чем больше ты производишь, чем больше делаешь, тем выше твоя эффективность.
-

-
- С таким подходом можно «зарезать на ужин курицу, несущую золотые яйца».
 - Затраты не следует путать с инвестициями.
 - Оплата аренды, электроэнергии, коммунальные платежи — **затраты**.
 - Создание и закрепление эффективной команды — это стратегическое приобретение компании.
 - Обучение участников проекта — **инвестиции**.
 - **Вложение в людей** — это увеличение числителя в формуле эффективности.
-

-
- Уход из компании всех профессионалов после проекта, выполненного по принципу «**любой ценой**», — затраты, причем очень тяжело восполняемые.
 - Нарастающая конкуренция указывает на совершенно четкий тренд в мировой экономике — персонал — это форма инвестиций, активов, которые нужно уметь наращивать, управлять и сохранять.
 - **Сегодня люди — это капитал.**
-

-
- Современное предприятие обязано относиться к своим работникам так же, как к своим лучшим клиентам.
 - Главный капитал современной компании — это **знания**.
 - Большая часть этих знаний неотъемлема от их носителя — **человека**.
 - Те предприятия, которые этого не поняли, не выживут потому, что не смогут быть эффективными.
-

-
- Сегодня эффективное предприятие — **это сервис.**
 - Предприятие, с одной стороны, предоставляет услуги и продукты своим клиентам, а с другой, — рабочие места для профессионального персонала.
 - Принципы «Одно предприятие на всю жизнь», «Работай продуктивно, а предприятие о тебе позаботится» — уходят в прошлое.
 - Посмотрите на рынок рабочей силы в ИТ — **правила устанавливают профессионалы.**
-

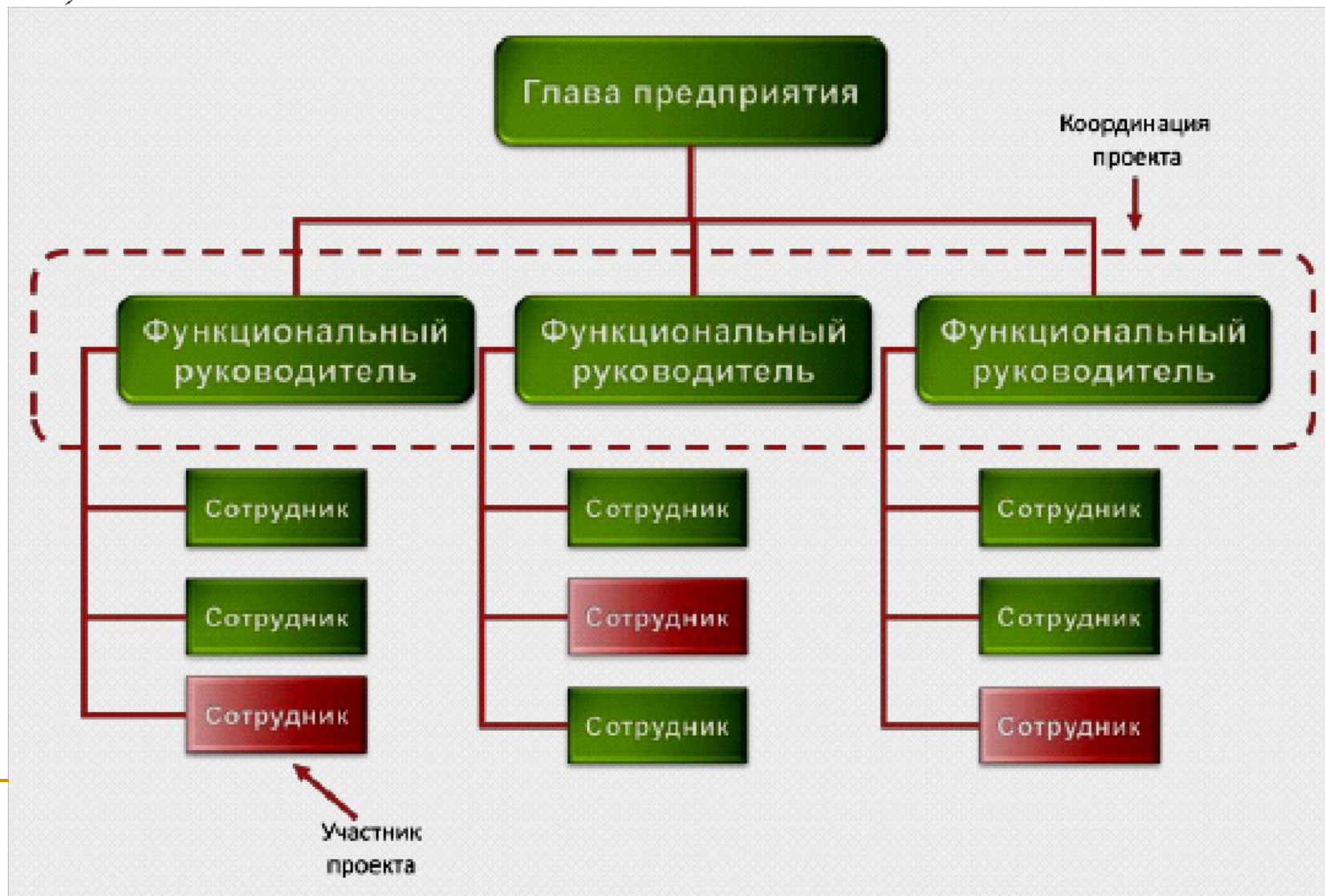
■ **Организационная структура** компании отражает ее внутреннее устройство, потоки управляющих воздействий, распределение труда и специфические особенности производства.

■ **Функциональная и проектная организации** — противоположные полюса, а **матричная организация** — промежуточные состояния.

■ Нет одной лучшей организационной структуры.

■ Нет смысла противопоставлять функциональные структуры и проектные организации.

Синоним функциональной структуры — иерархическая структура (рис. 5.3).



■ **Функциональная структура** имеет *следующие особенности:*

- • Сохраняется принцип единоначалия
 - • Понятные и стабильные условия работы
 - • Хорошо приспособлены для операционной деятельности.
 - • Специализация подразделений позволяет накапливать экспертизу.
 - • Затруднено принятие решений и коммуникации между исполнителями. Осуществляются только через руководство.
-

- Управление сконцентрировано и держится на компетенции высшего руководства
- Как правило, неэффективен контроль за ходом проекта (нет целостной картины)
- Функциональная структура предполагает многоуровневую иерархию.
- **Руководители функциональных подразделений** это начальники управлений, начальники подчиненных им служб, отделов, лабораторий, секторов, групп. У каждого начальника есть заместитель и, порой, не один.
- **Примеры:** министерства, ведомства, научные институты и предприятия советского периода.

■ На другом краю спектра организационных структур находится **проектная структура**



-
- В чисто **проектных организациях**:
 - Проект организуется как самостоятельное производственное подразделение.
 - Персонал на проект набирается по временным контрактам.
 - После завершения проекта персонал увольняется.
 - Медленный старт.
 - Опыт не аккумулируется.
 - Команды не сохраняются.
-

-
- **Проектные организации** не самые эффективные, но порой единственно возможные для выполнения проектов, которые физически удалены от исполняющей организации, например, строительство нового нефтепровода.
 - В разработке ПО наиболее распространена **матричная организация**.
 - Различают три вида матричной организационной структуры:
 - слабая,
 - сбалансированная
 - и сильная
-

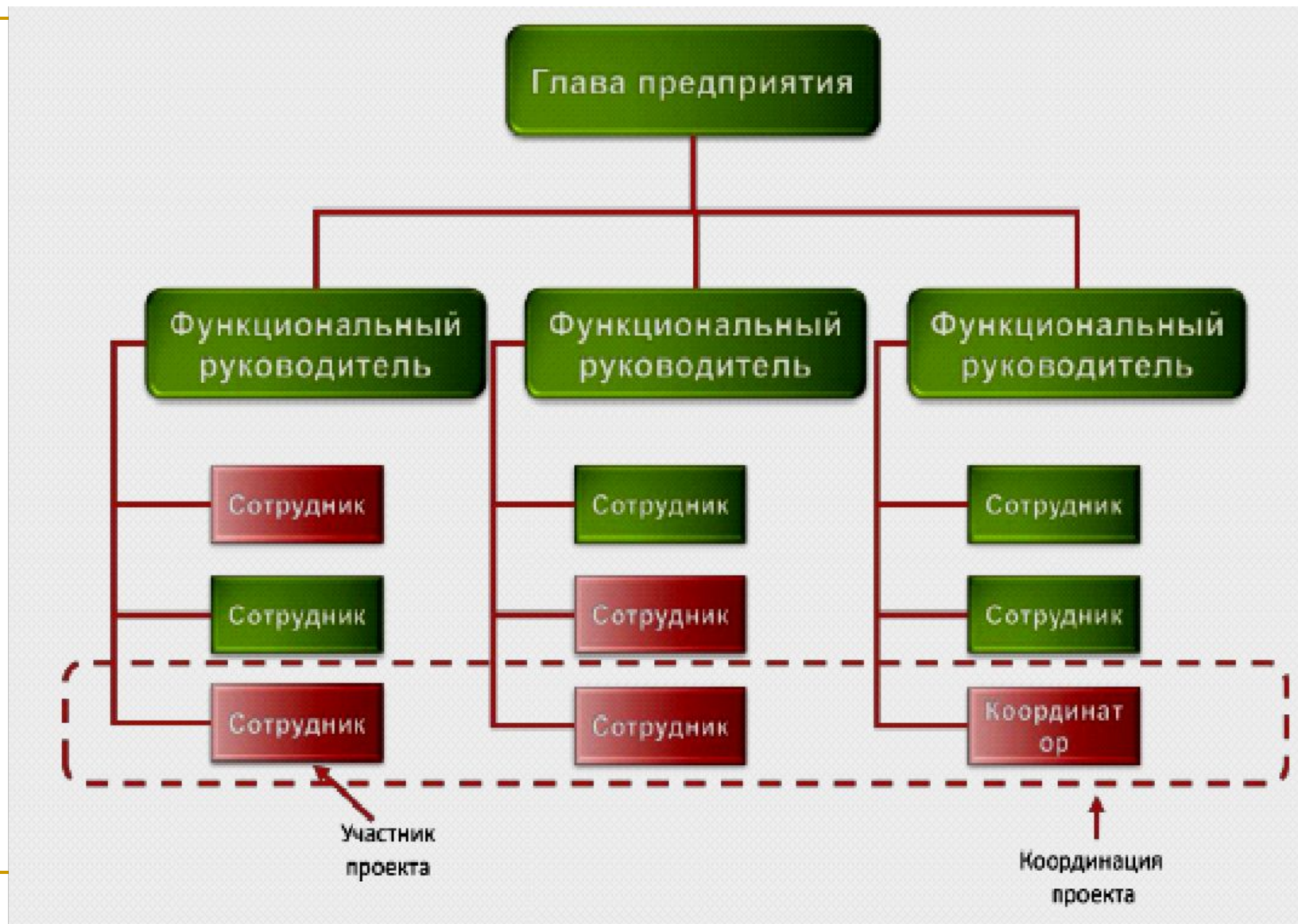


Рис. 5.5. Слабая матрица

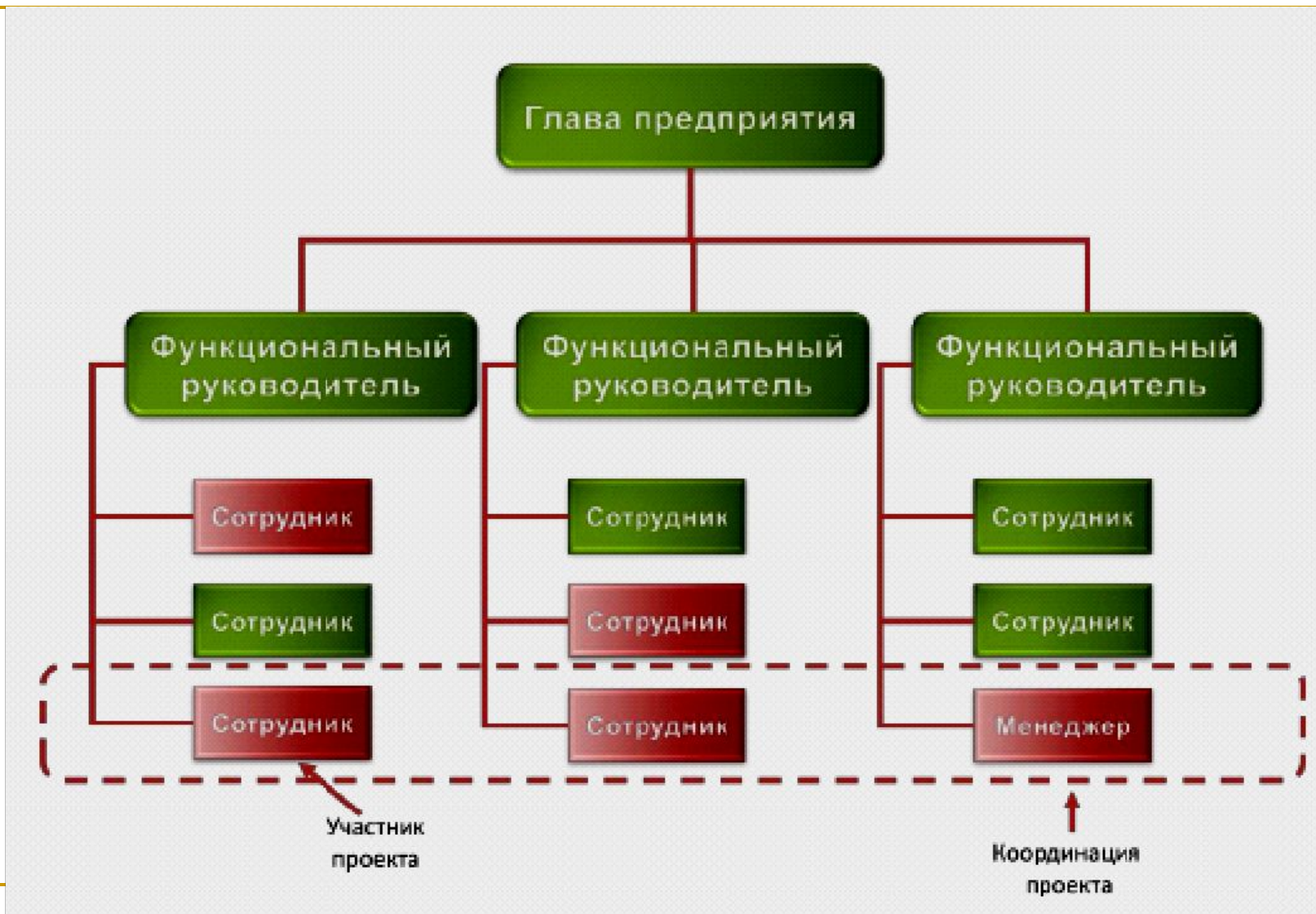


Рис. 5.6. Сбалансированная матрица



Рис. 5.7. Сильная матрица

-
- В компаниях, которые ориентированы в основном на заказную разработку ПО, функциональные подразделения чаще объединяются в соответствие с **используемыми информационными технологиями**.
 - Например, отдел разработки баз данных, отдел разработки J2EE-приложений, отдел веб-разработок, отделы тестирования, документирования и т.д.
-

-
- В ***слабой матрице*** роль и полномочия сотрудника, который **координирует проект, сильно ограничены.**
 - Реальное руководство проектом осуществляет один из **функциональных руководителей.**
 - Координатор проекта, его еще часто называют «трекер», помогает этому руководителю собирать информацию о статусе выполняемых проектных работ, учитывает затраты, составляет отчеты.
-

- В **сбалансированной матрице** появляется **менеджер проекта**, который реально управляет выделенными на проект ресурсами.
- Он планирует работы, распределяет задачи среди исполнителей, контролирует сроки и результаты, несет полную ответственность за достижение целей проекта, при соблюдении ограничений.
- В сбалансированных матрицах наиболее ярко проявляется проблема двойного подчинения.
- Руководитель функционального подразделения и менеджер проекта имеют примерно равное влияние на материальный и профессиональный рост разработчиков.

-
- В **сильной матрице** признается, что проектное управление является самостоятельной областью компетенции, в которой необходимо накапливать экспертизу и использовать общие ресурсы.
 - Поэтому в сильной матрице менеджеры проектов объединяются в самостоятельное функциональное подразделение — **офис управления проектами (ОУП)**.
 - ОУП разрабатывает корпоративные политики и стандарты в области проектного управления, планирует и осуществляет профессиональное развитие менеджеров.
-

■ Одной из особенностей *матричных структур* является то, что они становятся «плоскими», ***исчезает многоступенчатая иерархия.***

■ Предприятие, как правило, делится на функциональные отделы, в которых работают специалисты разных категорий, напрямую подчиняющиеся начальнику отдела.

■ Начальники лабораторий, секторов, групп упраздняются за ненужностью.

■ В матричных структурах **роль начальника функционального подразделения** в производственном процессе заметно снижается, по сравнению с функциональными структурами.

-
- В его компетенции остаются вопросы стратегического развития функционального направления, планирование и развитие карьеры сотрудников, вопросы материально-технического обеспечения работ.
 - Следует учитывать, что такое перераспределение полномочий и ответственности от функциональных руководителей к менеджерам проектов часто служит **источником конфликтов** в компаниях при их переходе от функциональной структуры к матричной.
-

-
- Каждый проект разработки ПО имеет свою **организационную структуру**, которая определяет распределение ответственности и полномочий среди участников проекта, а также обязанностей и отношений отчетности.
 - Чем меньше проект, тем больше ролей придется совмещать **одному исполнителю**.
-

■ Роли и ответственности участников типового проекта разработки ПО можно условно разделить на **пять групп**:

□ 1) **Анализ**. Извлечение, документирование и сопровождение требований к продукту.

□ 2) **Управление**. Определение и управление производственными процессами.

□ 3) **Производство**. Проектирование и разработка ПО.

□ 4) **Тестирование**. Тестирование ПО.

□ 5) **Обеспечение**. Производство дополнительных продуктов и услуг.

■ **Группа анализа** включает следующие *роли*:

- **Бизнес-аналитик**. Построение модели предметной области (онтологии).

- **Бизнес-архитектор**. Разрабатывает бизнес-концепцию системы. Определяет общее видение продукта, его интерфейсы, поведение и ограничения.

- **Системный аналитик**. Отвечает за перевод требований к продукту в функциональные требования к ПО.

- **Специалист по требованиям**. Документирование и сопровождение требований к продукту.

- **Менеджер продукта** (функциональный заказчик). Представляет в проекте интересы пользователей продукта.

■ **Группа управления** состоит из следующих **ролей**:

• **Руководитель проекта**. Отвечает за достижение целей проекта при заданных ограничениях (по срокам, бюджету и содержанию), осуществляет операционное управление проектом и выделенными ресурсами.

• **Куратор проекта**. Оценка планов и исполнения проекта. Выделение ресурсов.

• **Системный архитектор**. Разработка технической концепции системы. Принятие ключевых проектных решений относительно внутреннего устройства программной системы и её технических интерфейсов.

• **Руководитель группы тестирования**. Определение целей и стратегии тестирования, управление тестированием.

• **Ответственный за управление изменениями, конфигурациями, за сборку и поставку программного продукта.**

■ В ***производственную группу*** входят:

- ***Проектировщик***. Проектирование компонентов и подсистем в соответствии с общей архитектурой, разработка архитектурно значимых модулей.
 - ***Проектировщик базы данных***.
 - ***Проектировщик интерфейса пользователя***
 - ***Разработчик***. Проектирование, реализация и отладка отдельных модулей системы.
-

-
- В большом проекте может быть **несколько производственных групп**, ответственных за отдельные подсистемы.
 - Как правило, проектировщик выполняет роль лидера группы и управляет своим подпроектом или пакетом работ.
 - Стоит не забывать, что **руководитель проекта делегирует полномочия**, но не ответственность.
-

-
- **Группа тестирования** в проекте состоит из следующих ролей:
 - **Проектировщик тестов**. Разработка тестовых сценариев.
 - **Разработчик автоматизированных тестов**.
 - **Тестировщик**. Тестирование продукта. Анализ и документирование результатов.
-

■ Участники **группы обеспечения**, как правило, не входят в команду проекта. Они выполняют работы в рамках своей процессной деятельности. К группе обеспечения можно отнести следующие роли:

- Технический писатель.
 - Переводчик.
 - Дизайнер графического интерфейса.
 - Разработчик учебных курсов, тренер.
 - Участник рецензирования.
 - Продажи и маркетинг.
 - Системный администратор.
 - Технолог и др.
-

■ В зависимости от масштаба проекта **одну роль** могут исполнять **несколько человек**.

■ *Например*, разработчики, тестировщики, технические писатели.

■ **Некоторые роли** всегда должен исполнять **только один человек**.

■ *Например*, Руководитель проекта, Системный архитектор.

■ Один человек может исполнять несколько ролей.

■ Возможны следующие **совмещения ролей**:

□ • Руководитель проекта + системный аналитик (+ системный архитектор)

□ • Системный архитектор + разработчик

□ • Системный аналитик + проектировщик тестов (+ технический писатель)

□ • Системный аналитик + проектировщик интерфейса пользователя

□ • Ответственный за управление конфигурациями + ответственный за сборку и поставку (+ разработчик)

■ **Крайне нежелательно совмещать следующие роли:**

- ❑ Разработчик + руководитель проекта
 - ❑ Разработчик + системный аналитик.
 - ❑ Разработчик + проектировщик интерфейсов пользователя.
 - ❑ Разработчик + тестировщик
-

-
- **Не раз приходилось наблюдать**, как в критические периоды проекта его менеджер-разработчик с увлечением правит очередные баги, а проектная команда в полном составе стоит у него за спиной и наблюдает за этим процессом.
 - Это **плохой пример руководства проектом**.
 - Программисты любят и умеют программировать.
 - Пусть они этим и занимаются.
 - Не стоит загружать программистов несвойственной для них работой.
-

-
- В каждом проекте разработки программного продукта **много других работ**: бизнес-анализ, проектирование эргономики, графический дизайн, разработка пользовательской документации.
 - Эти работы с программированием **не имеют ничего общего**.
 - Для них требуются совершенно другая квалификация и другой склад мышления.
 - При кустарном производстве программ эти задачи, как правило, поручаются программистам, которые это делать не умеют и не любят.
 - Получается обычно плохо, да еще и дорого.

-
- В силу своей **интроверсии**, граничащей с аутизмом, программист просто не в состоянии увидеть свою программу чужими глазами — **глазами пользователей**
 - Никто уже не хочет работать с программами с технологической парадигмой навороченного пользовательского интерфейса — кустарным творением программистов — когда для того чтобы работать с системой, надо обязательно знать, как она устроена.
 - Интроверсия – (лат. intro - внутрь, versio - поворачивать, обращать) – ~~обращенность сознания человека к самому себе; поглощенность собственными проблем...~~

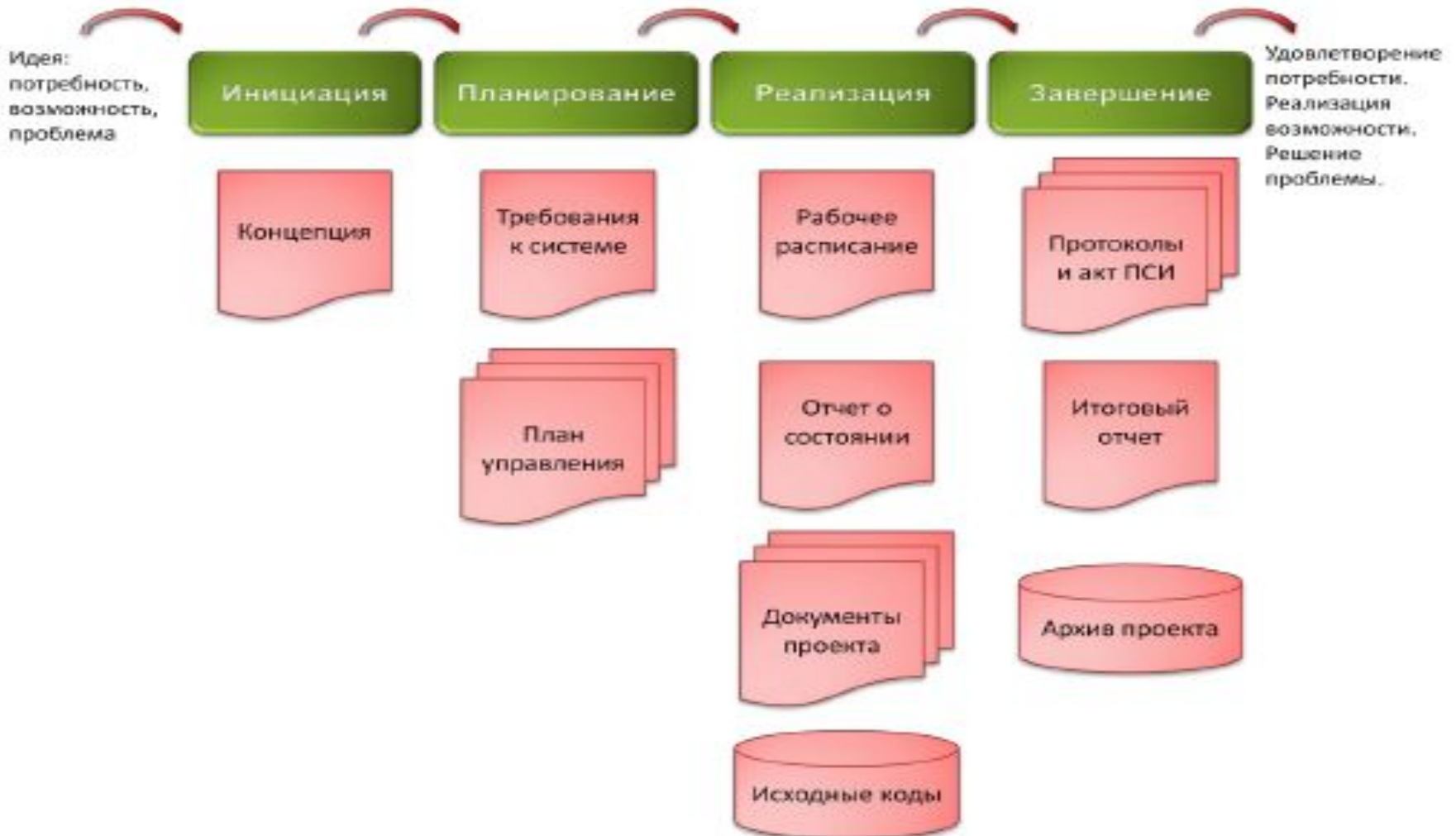
-
- Это **типичное творение** программиста, которому гораздо важнее видеть, как работает его программа, чем разбираться в том, **что она делает для пользователя.**
 - Поэтому, необходимо привлекать в проектную команду **бизнес-аналитиков, эргономистов, художников-дизайнеров, документалистов.**
 - **Разделение труда и специализация** — залог перехода от кустарного производства к более эффективному промышленному производству.
-

-
- Из профессиональных программистов получаются отличные **тестировщики**.
 - Однако, совмещать одновременно роли программиста и тестировщика — **плохая практика**.
 - Хороший программист убежден, что он пишет программы правильно и ему психологически тяжело допустить, что где-то в его коде может быть ошибка.
 - ***А ошибки есть всегда!***
-

- Организационная структура проекта обязательно должна включать в себя **эффективную систему отчетности**, оценки хода выполнения проекта и систему принятия решений.
- Можно рекомендовать **еженедельные собрания** по статусу проекта, на которых анализируются риски, оцениваются результаты, достигнутые на предыдущей неделе, и уточняются задачи на новый период.
- В модели Scrum рекомендуются **ежедневные совещания** по состоянию работ — «Stand Up Meeting», но это, наверно, применимо, скорее, для небольших рабочих групп от 3 до 5 разработчиков.

-
- Важно помнить, что организационная структура проекта — **«живой» организм**.
 - Она начинает складываться на стадии планирования и может **меняться в ходе проекта**.
 - Нестабильность организационной структуры (частые замены исполнителей) — **серьезная проблема в управлении сложными программными проектами**, поскольку существует время вхождения в контекст проекта, которое может измеряться месяцами.
-

Каждый проект разработки ПО состоит из 4-х фаз



- На фазе **инициации** проекта необходимо понять, что и зачем мы будем делать — разработать концепцию проекта.
- Фаза **планирования** определяет, как мы будем это делать.
- На фазе **реализации** происходит материализация наших идей в виде документированного и протестированного программного продукта.
- И на фазе **завершения** мы должны подтвердить, что мы разработали именно тот продукт, который задумали, и провести приемо-сдаточные испытания продукта на предмет соответствия его свойств, определенным ранее требованиям.

■ Как правило, **редкий проект выполняется в соответствии с первоначальными планами**, поэтому важным элементом фазы завершения является «обратная связь»:

□ анализ причин расхождения и усвоение уроков на будущее.

■ **Помним, что управляющая система без обратной связи не может быть устойчивой.**

-
- Завершая обзор управления проектами «с высоты птичьего полета», необходимо упомянуть еще об одной особенности проекта по сравнению с операционной деятельностью.
 - Если в операционной деятельности ресурсы расходуются более-менее равномерно по времени, то в проектном управлении **расходование ресурсов в единицу времени имеет явно выраженное колоколообразное распределение** (рис. 5.9).
-

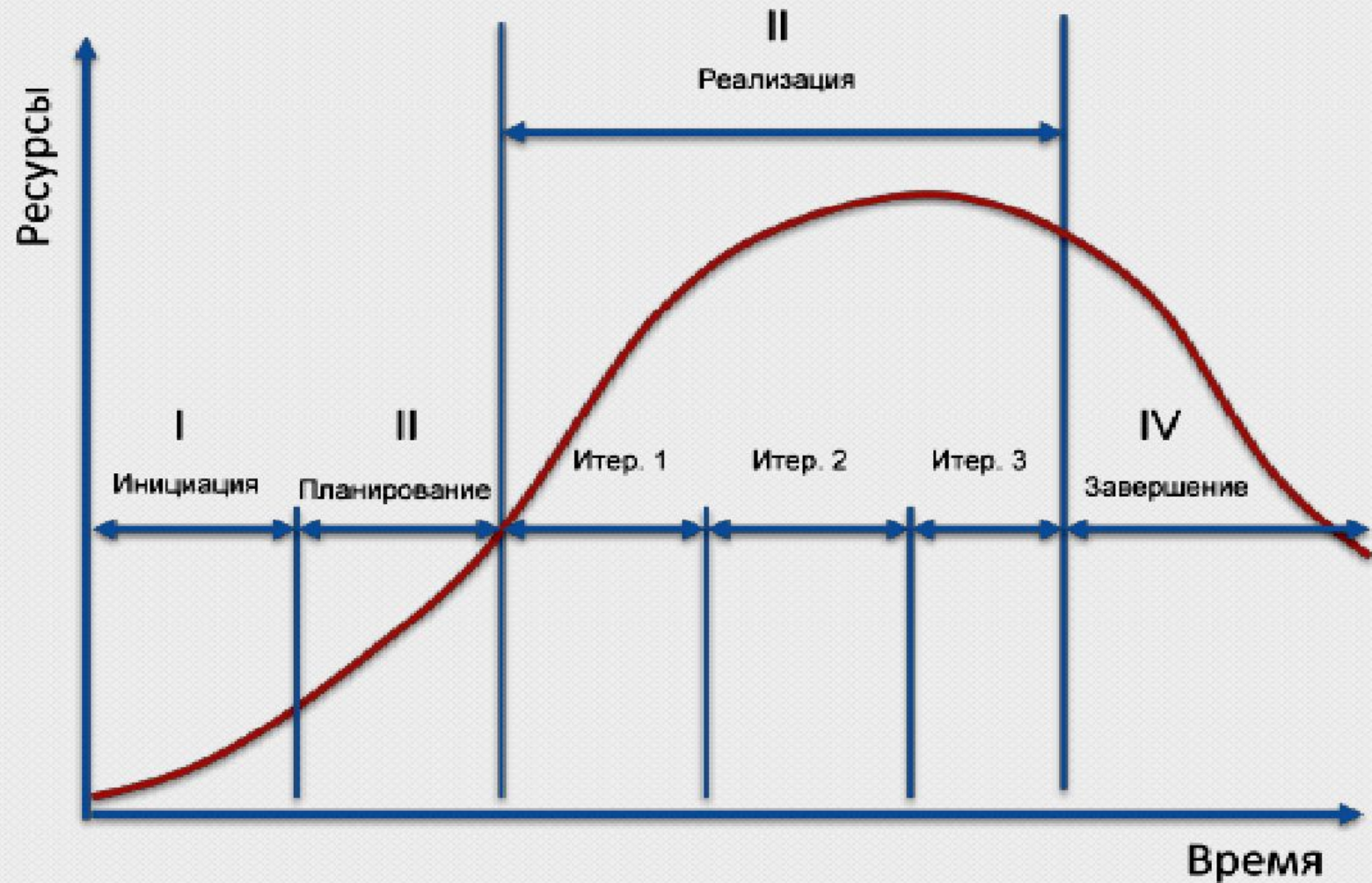


Рис. 5.9. Распределение ресурсов по фазам проекта

-
- Проект часто начинается с **идеи**, которая появляется у одного человека.
 - Постепенно, по мере **формулирования, анализа и оценки этой идеи**, привлекаются дополнительные специалисты.
 - Еще больше участников требуется на фазе **планирования проекта**. Пик потребления ресурсов приходится на **фазу реализации**.
 - В современных моделях разработки ПО реализация осуществляется на основе сочетания **итеративного** и **инкрементального** подходов.
-

■ **Итеративность** предполагает, что требования к системе и ее архитектура прорабатываются не один раз, а **постепенно уточняются от итерации к итерации.**

■ Это означает, что на каждой итерации происходит **полный цикл процессов разработки:**

□ уточнение требований,

□ проектирование,

□ кодирование,

□ тестирование

□ и документирование.

-
- **Инкрементальность** состоит в том, что
 - результатом каждой итерации является версия ПО, которая реализует часть функциональности будущего программного продукта
 - и может быть введена в тестовую или опытную эксплуатацию,
 - а также оценена заказчиком и будущими пользователями.
 - Это означает, что после каждой итерации **происходит прирост требуемого функционала**, а нереализованных функций будущего продукта остается все меньше.

-
- Сочетание **итеративности и инкрементальности** обеспечивает эффективность разработки и существенное снижение рисков по ходу проекта.
 - На последней фазе происходит постепенное высвобождение участников проектной команды.
 - Следует помнить, что проект должен иметь **четкое окончание во времени**, после которого все работы по проекту закрываются, и на проект перестают тратиться ресурсы.
 - Не должно оставаться **«зависших» работ**.
-

-
- **Проект** — это средство стратегического развития.
 - **Цель** — описание того, что мы хотим достичь.
 - **Стратегия** — констатация того, каким образом мы собираемся эти цели достигать.
 - Проекты преобразуют **стратегии в действия**, а **цели в реальность**.
-

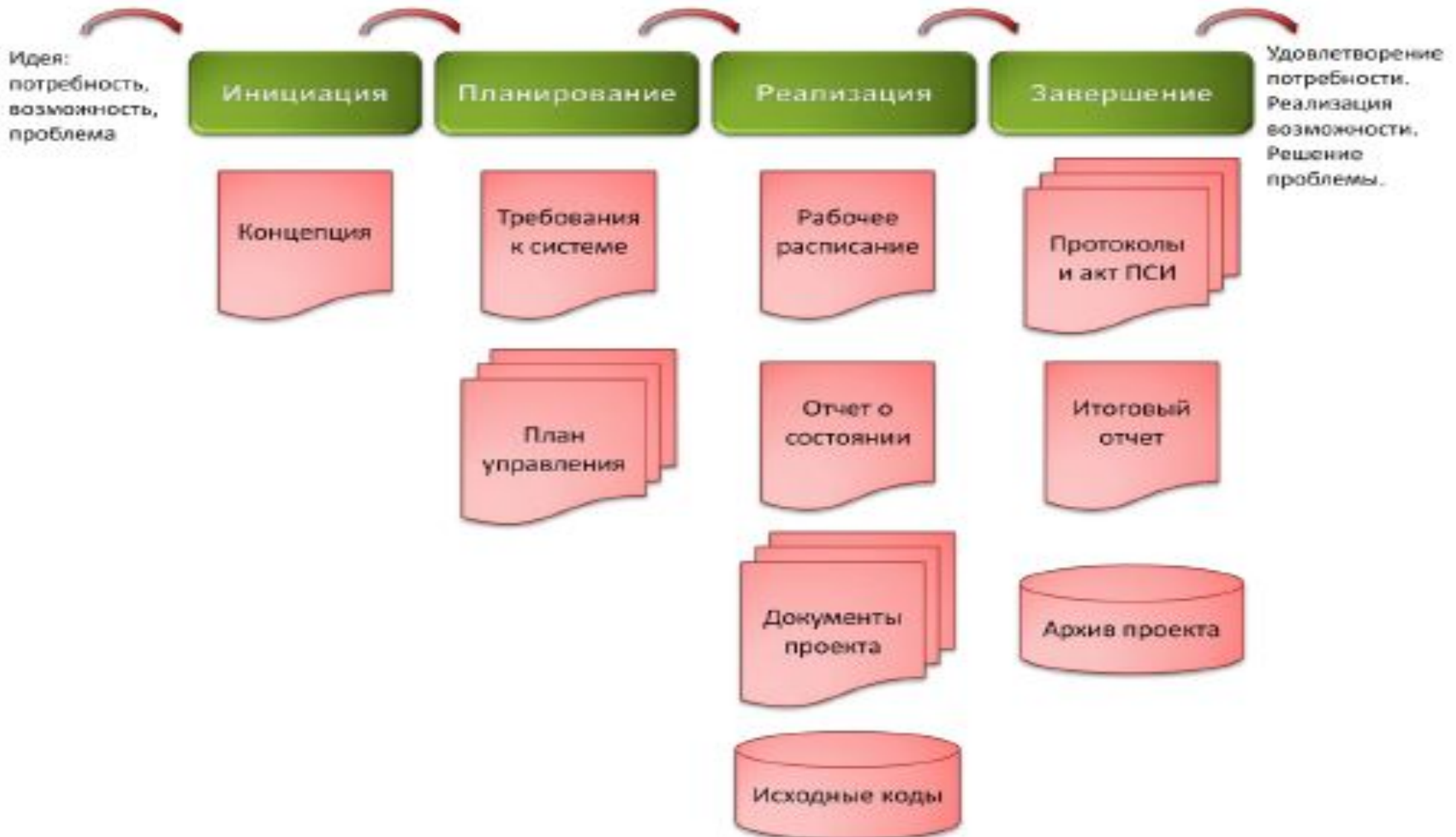
Участников типового проекта разработки ПО

можно условно разделить на пять **групп ролей**:

- **Анализ.** Извлечение, документирование и сопровождение требований к продукту.
 - **Управление.** Определение и управление производственными процессами.
 - **Производство.** Проектирование и разработка ПО.
 - **Тестирование.** Тестирование ПО.
 - **Обеспечение.** Производство дополнительных продуктов и услуг.
-

ВЫВОДЫ

- Каждый проект разработки ПО состоит из 4-х фаз



■ У программного проекта имеется четыре фактора, которые определяют его успешность:

- Выполнен в соответствии со **спецификациями**.
 - Выполнен в **срок**.
 - Выполнен в **пределах бюджета**.
 - Каждый участник команды уходил с работы в **18:00** с чувством успеха.
-

■ В современных моделях разработки ПО реализация осуществляется на основе сочетания **итеративного** и **инкрементального** подходов.

■ **Итеративность** предполагает, что требования к системе и ее архитектура прорабатываются не один раз, а **постепенно уточняются от итерации к итерации**.

■ Это означает, что на каждой итерации происходит **полный цикл процессов разработки**:

□ уточнение требований,

□ проектирование,

□ кодирование,

□ тестирование

□ и документирование.

ВЫВОДЫ

- **Инкрементальность** состоит в том, что
 - результатом каждой итерации является версия ПО, которая реализует часть функциональности будущего программного продукта
 - и может быть введена в тестовую или опытную эксплуатацию,
 - а также оценена заказчиком и будущими пользователями.
- Это означает, что после каждой итерации **происходит прирост требуемого функционала**, а нереализованных функций будущего продукта остается все меньше.

-
- 1. Чем отличается проектная деятельность от операционной?
 - 2. Проект как средство стратегического развития: от цели до реализации.
 - 3. Что означает информационное или постиндустриальное общество?
 - 4. Критерии (факторы) успешности программного проекта?
 - 5. Особенности функциональной организационной структуры предприятия.
 - 6. Виды проектной организационной структуры предприятия.
 - 7. Роли и ответственность участников проектной команды.
 - 8. Какие роли можно совмещать, а какие нет в проектной команде?
 - 9. Жизненный цикл проекта. Фазы и продукты.
 - 10. Итеративный и инкрементальные подходы разработки программного обеспечения.

Лекция

**УПРАВЛЕНИЕ ПРОЕКТАМИ.
ОПРЕДЕЛЕНИЯ И
КОНЦЕПЦИИ**

Король Иван Андреевич
