
Лекция

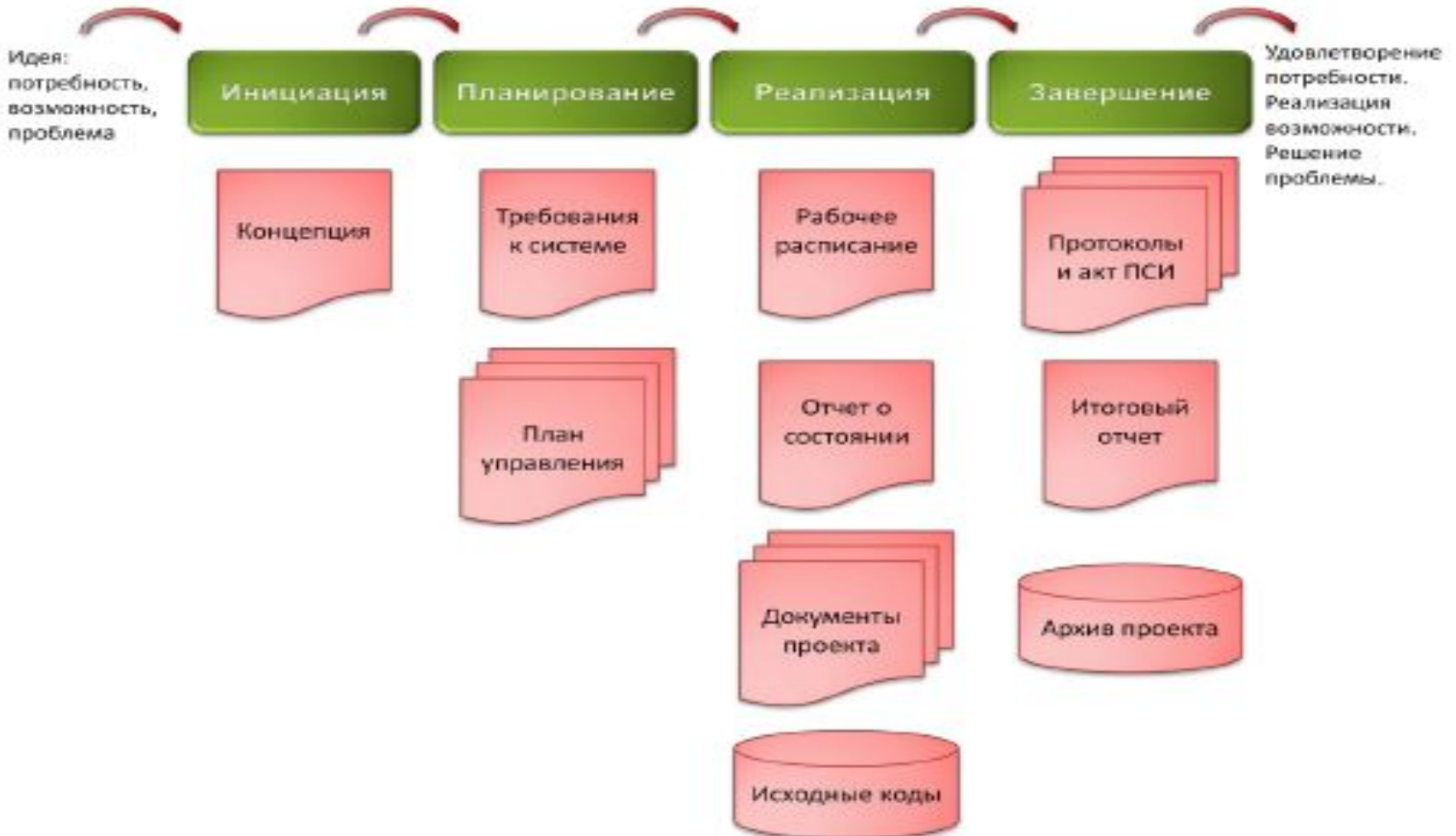
ПЛАНИРОВАНИЕ ПРОЕКТА

Король Иван Андреевич

Вопросы

- 1. Уточнение содержания и состава работ
 - 2. Планирование управления содержанием
 - 3. Планирование организационной структуры
 - 4. Планирование управления конфигурациями
 - 5. Планирование управления качеством
 - 6. Базовое расписание проекта
 - 7. Выводы
 - 8. Контрольные вопросы
-

Каждый проект разработки ПО состоит из 4-х фаз



■ **«Если не получается проглотить слона целиком, то его надо порезать на отбивные».**

■ Человечество пока не придумало ничего более эффективного для решения сложной задачи, чем анализ и ее **декомпозиция** на более простые подзадачи, которые, в свою очередь, могут быть разделены на еще более простые подзадачи и т.д.

■ Получается некоторая **иерархическая структура**, дерево, в корне которого находится проект, а на листьях элементарные задачи или работы, которые надо выполнить, чтобы завершить проект в условиях заданных ограничений.

-
- Согласно [1]: «PMBOK. Руководство к Своду знаний по управлению проектами», 3-е изд., PMI, 2004.
 - **Иерархическая структура работ (ИСР)** (*Work Breakdown Structure, WBS*) — ориентированная на результат иерархическая декомпозиция работ, выполняемых командой проекта для достижения целей проекта и необходимых результатов.
 - С ее помощью **структурируется и определяется** все содержание проекта.
 - Каждый следующий **уровень иерархии** отражает более детальное определение элементов проекта.
-

■ Основой для разработки ИСР служит **концепция проекта**, которая определяет *продукты проекта* и их *основные характеристики*.

■ ИСР обеспечивает выявление **всех работ**, необходимых для достижения целей проекта.

■ Многие проекты проваливаются не от того, что у них нет плана, а от того что в этом плане **забыты важные работы**, например тестирование и исправление ошибок, и продукты проекта, например пользовательская документация.

■ Поэтому, если ИСР составлена **корректно**, то ~~любая работа, которая в нее не вошла не может~~ считаться работой по проекту.

-
- ИСР делит проект на **подпроекты, пакеты работ, подпакеты**.
 - Каждый следующий уровень декомпозиции обеспечивает последовательную детализацию содержания проекта, что позволяет производить **оценку сроков и объемов работ**.
 - ИСР должна включать все **промежуточные и конечные продукты**.
-

■ Выполнять декомпозицию работ проекта можно по-разному.

■ Например, ГОСТ 19.102-77 предусматривает **каскадный подход** и определяет следующие стадии разработки программной системы:

- Техническое задание
 - Эскизный проект
 - Технический проект
 - Рабочий проект
 - Внедрение
-

-
- Если следовать этому стандарту, то на первом уровне ИСР должны находиться именно эти проектные продукты.
 - Если бы пришлось разрабатывать АСУ для управления ядерным реактором или пилотируемым космическим аппаратом, то именно так и следовало поступать.
 - Однако в **коммерческой разработке ПО** такой подход не эффективен.
 - Как мы уже говорили, современный процесс разработки коммерческого ПО должен быть **инкрементальным**.
-

- Это означает, что на верхнем уровне декомпозиции нашего проекта должны находиться **продукты** проекта, а на следующем уровне — **компоненты**, из которых эти продукты состоят.
- Компоненты далее могут быть декомпозированы на «**фичи**» — функции, которые они должны реализовывать.
- Выделение компонентов, составляющих программный продукт, это элемент **высокоуровневого проектирования**, которое мы должны выполнить на фазе **планирования проекта**, ~~не дожидаясь проработки всех функциональных требований к разрабатываемому ПО.~~

-
- **Компонентами** могут быть как **прикладные подсистемы**, так и **инфраструктурные** или **ядерные**, например, подсистема логирования, безопасности, библиотека визуальных компонентов GUI.
 - При составлении базового плана работ не стоит стремиться максимально детализировать все работы.
 - **ИСР** не должна содержать слишком много **уровней**, достаточно **3-5**.
-

■ Например, ИСР нашего проекта-примера разработки «Автоматизированной системы продажи документации» может выглядеть следующим образом

■ 1. Проект разработки «Автоматизированной системы продажи документации»

■ **1.1. Подготовка технического задания на автоматизацию**

- 1.1.1. Проведение аналитического обследования
 - 1.1.2. Разработка функциональных требований
 - 1.1.3. Разработка требований к базовому ПО
 - 1.1.4. Разработка требований к оборудованию и к операционно-системному ПО
 - 1.1.5. Согласование и утверждение ТЗ
 - **1.1.6. ТЗ утверждено**
-

■ 1.2. Поставка и монтаж оборудования

- 1.2.1. Разработка спецификации на оборудование
- 1.2.2. Закупка и поставка оборудования
- 1.2.3. Монтаж оборудования
- 1.2.4. Установка и настройка операционно-системного ПО
- **1.2.5. Монтаж оборудования завершен**

■ 1.3. Поставка и установка базового ПО

- 1.3.1 .Разработка спецификаций на базовое ПО
 - 1.3.2. Закупка базового ПО
 - 1.3.3. Развертывание и настройка базового ПО
 - *1.3.4. Базовое ПО установлено у заказчика*
-

■ 1.4. Разработка и тестирование прикладного ПО

- 1.4.1. Разработка спецификаций на прикладное ПО
- 1.4.2. Установка и конфигурирование рабочей среды
- 1.4.3. **Проектирование и разработка ПО**
 - 1.4.3.1. Авторизация и аутентификация пользователей.
 - 1.4.3.2. **Разработка подсистемы заказа документации**
 - 1.4.3.2.1. Просмотр каталога продуктов.
 - 1.4.3.2.2. Поиск продуктов по каталогу.
 - 1.4.3.2.3. Заказ выбранных продуктов.
 - 1.4.3.2.4. Просмотр информации о статусе заказа.
 - 1.4.3.2.5. Информирование клиента об изменении статуса заказа.
 - 1.4.3.2.6. *Подсистема заказа документации передана в тестовую эксплуатацию (на серверах разработчика).*

- 1.4.3.3. **Разработка подсистемы обработки заказов**
 - 1.4.3.3.1. Просмотр и обработка заказов исполнителями из службы продаж.
 - 1.4.3.3.2. Просмотр статистики поступления и обработки заказов за период.
 - 1.4.3.3.3. *Подсистема обработки заказов передана в тестовую эксплуатацию на оборудовании Заказчика*
- 1.4.3.4. **Разработка подсистемы сопровождения каталога**
 - 1.4.3.4.1. Подготовка и сопровождение каталога продукции.
 - 1.4.3.5. Исправление ошибок
- **1.4.4. Тестирование ПО**
 - 1.4.4.1. Раунд 1
 - 1.4.4.2. Раунд 2
 - 1.4.4.3. Раунд 3
 - 1.4.4.4. Выходное тестирование
 - 1.4.5. Документирование прикладного ПО

■ 1.5. Обучение пользователей

- 1.5.1 .Подготовка учебных курсов
- 1.5.2. Обучение сотрудников Отдела 123
- 1.5.3. Обучение руководства ОАО XYZ
- 1.5.4. Обучение администраторов системы

■ 1.6. Ввод в опытную эксплуатацию

- 1.6.1. Развертывание и настройка прикладного ПО
- 1.6.2. Проведение приемо-сдаточных испытаний
- 1.6.3. *Акт передачи системы в опытную эксплуатацию*

■ 1.7. Сопровождение системы в период опытной эксплуатации

■ 1.8. Система передана в промышленную экспл.

- Должна быть установлена **персональная ответственность** за все части проекта (подпроекты и пакеты работ).
- Для каждого пакета работ должен быть четко определен **результат на выходе**.
- Работы и оценки проекта должны быть **согласованы** с ключевыми участниками команды, **руководством компании-исполнителя** и, при необходимости, с **заказчиком**.
- В результате согласования члены команды принимают на себя обязательства по **реализации** проекта, а руководство принимает на себя обязательства по **обеспечению** проекта ресурсами

Иерархическая структура работ

- ИСР является одним из **основных инструментов** (средств) в механизме управления проектом, с помощью которого измеряется степень достижения результатов проекта.
- Важнейшая ее функция это обеспечить **консистентное** (непротиворечивое, совместимое) **представление** всех участников проекта относительно того, как будет делаться проект.
- В последующем базовый план будет служить **ориентиром** для сравнения с текущим исполнением проекта и выявления **отклонений** для целей управления.

Планирование управления содержанием

■ Управление изменениями

■ Одна из распространенных «**болезней**» программных проектов называется **фичеризм** (featurism), часто ползучий фичеризм (creeping featurism).

■ Это, когда к изначально спроектированной **будке** для любимой собаки сначала пристраивают **сарайчик** для хранения садового инвентаря, а потом и **домик** в несколько этажей для ее хозяина.

■ И все это пытаются построить на одном и том же **фундаменте** и из тех же самых материалов.

■ Эта болезнь стала причиной **летального исхода** многих проектов разработки ПО.

Планирование управления содержанием

- Поэтому, сразу, как только удалось стабилизировать и согласовать ИСР, необходимо разработать **план управления содержанием проекта**. Для этого следует:
 - Определить источники запросов на изменение.
 - Установить порядок анализа, оценки и утверждения/отклонения изменения содержания.
 - Определить порядок документирования изменений содержания.
 - Определить порядок информирования об изменении содержания.

Планирование управления содержанием

■ Первая задача, которую необходимо решить при анализе запроса на изменения — **выявить объекты изменений**:

- требования,
- архитектура,
- структуры данных,
- исходные коды,
- сценарии тестирования,
- пользовательская документация, проч.

Планирование управления содержанием

- Затем требуется **спроектировать и детально описать изменения** во всех выявленных объектах
- И наконец, следует **оценить затраты** на внесение изменений, тестирование изменений и регрессионное тестирование продукта и их влияние на сроки проекта.
- Эта работа, которая потребует **затрат рабочего времени** и порой значительных разных **специалистов**: аналитиков, проектировщиков, разработчиков, тестировщиков, наконец, менеджера проекта.
- Поэтому эта работа должна **обязательно** быть учтена в плане.

Планирование организационной структуры

- **Организационная структура** это согласованное и утвержденное распределение ролей, обязанностей и целей деятельности ключевых участников проекта.

- Она в обязательном порядке должна включать в себя:

- **систему рабочих взаимоотношений** между рабочими группами проекта,
- **систему отчетности**, оценки хода выполнения проекта
- **и систему принятия решений.**

Планирование организационной структуры

- Следует помнить, что организационная структура проекта — «**ЖИВОЙ**» организм.
- Она начинает складываться на стадии планирования и должна меняться по ходу проекта.
- **И еще.** **Нестабильность организационной структуры** — частая смена исполнителей — может стать серьезной проблемой в управлении проектом, поскольку, **существует цена замены**, которая определяется временем вхождения нового участника в контекст проекта.

Планирование управления конфигурациями

- **Конфигурационное управление** один из важных процессов производства программного обеспечения.
- Мы будем говорить только о том, что эта работа должна быть **спланирована**.
- План проекта должен включать в себя работы по обеспечению **единого хранилища** всей проектной документации и разрабатываемого программного кода, **обеспечению сохранности** и **восстановление** проектной информации после сбоя.

Планирование управления конфигурациями

- Работы по **настройке рабочих станций и серверов**, используемых участниками проектной команды, тоже должны войти в план.
- Кроме этого в плане должны содержаться работы, необходимые для организации **сборки промежуточных выпусков** системы, а также ее **конечного варианта**.
- Эти работы, как правило, выполняет один человек — **инженер по конфигурациям**. Если проект небольшой, то эта роль может быть дополнительной для, например, менеджера проекта.
- «**Размазывать**» эту работу на всех участников проекта, во-первых, неэффективно.

Планирование управления конфигурациями

- Установка и конфигурирование среды разработки, например, баз данных и серверов приложений, требует определенных **компетенций и знаний** особенностей конкретных версий продуктов.
- Если эти навыки придется **осваивать** всем разработчикам, то на это уйдет слишком много рабочего времени.
- Во-вторых, «размазывание» работ по управлению конфигурациями может привести к **коллективной безответственности**, когда никто не знает, от чего не собирается проект и как откатиться к консистентной версии.

Планирование управления конфигурациями

- Управление конфигурациями может **множественно усложниться**, если проектной команде параллельно с разработкой новой функциональности продукта приходится **поддерживать несколько релизов этого продукта**, которые были установлены ранее у разных клиентов.
- Все эти работы **должны быть учтены** в плане проекта.

Планирование управления качеством

- **Обеспечение качества** еще одна из базовых областей знаний в программной инженерии.
- Относительно того, что такое качество ПО и как его эффективно обеспечивать, можно рассуждать очень и очень долго.
- Необходимо руководствоваться **утверждением**, что **обеспечение качества** это важная работа,
 - которая должна быть **спланирована** заранее
 - и выполняться по ходу всего программного проекта,
 - а не только во время приемо-сдаточных испытаний.

Планирование управления качеством

- При планировании этой работы необходимо понимать, что продукт проекта не должен обладать **наивысшим возможным качеством**, которое недостижимо за конечное время.

- Необходимое качество продукта **определяется требованиями** к нему.

- **И еще.** Основная задача обеспечения качества это не поиск ошибок в готовом продукте (выходной контроль) а **их предупреждение** в процессе производства.

Планирование управления качеством

■ **План управления качеством** должен включать в себя следующие работы:

- Объективную **проверку соответствия программных продуктов и технологических операций** применяемым стандартам, процедурам и требованиям.
- **Определение отклонений по качеству**, выявление их причин, применение мер по их устранению, а также контроль исполнения принятых мер и их эффективности.
- Представление высшему руководству **независимой информации о несоответствиях**, не устраняемых на уровне проекта.

Планирование управления качеством

■ Помимо перечисленных разделов **план проекта** должен включать еще:

- План управления рисками
 - Оценку трудоемкости и сроков работ
-

Базовое расписание проекта

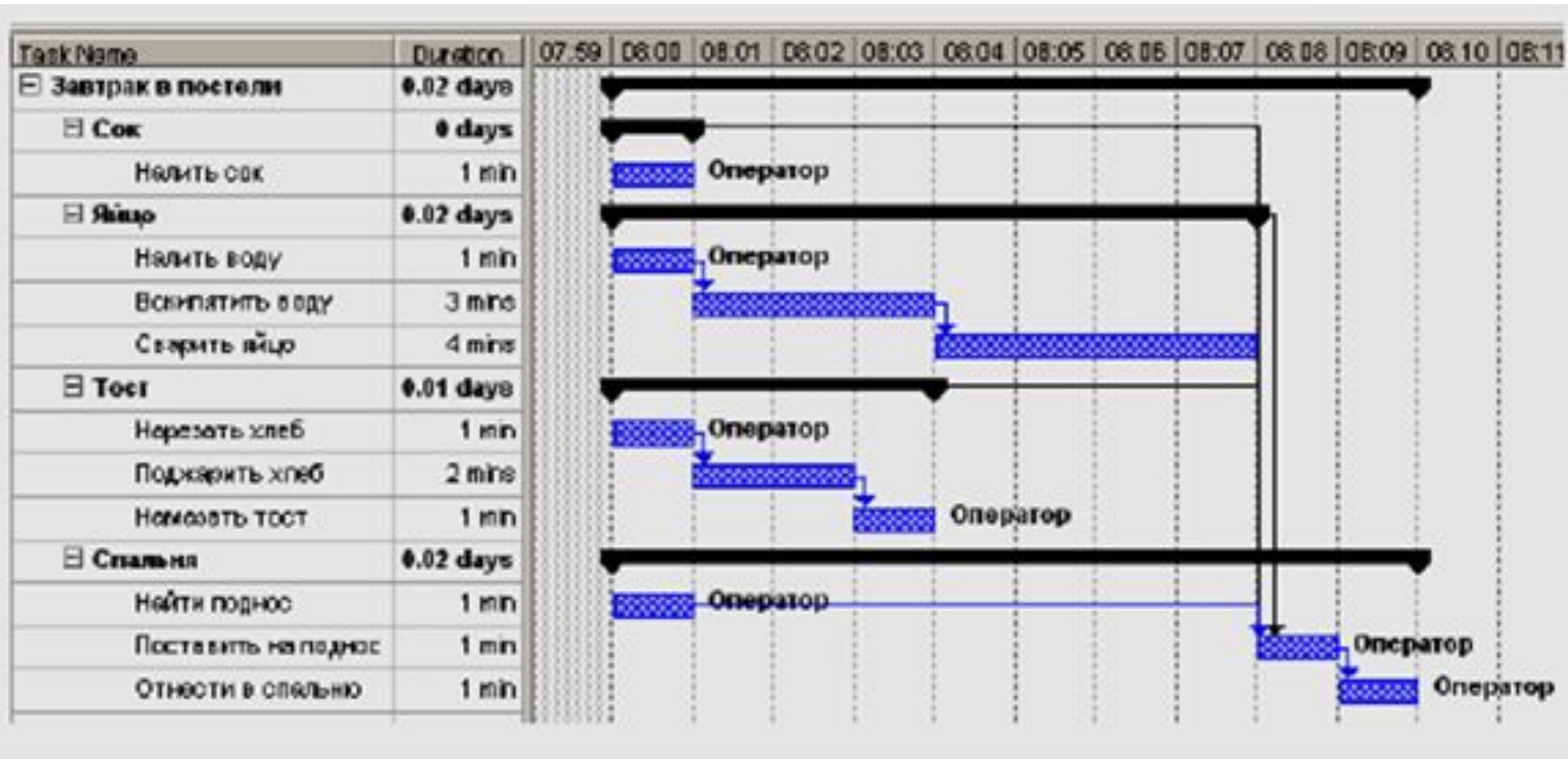
- После определения трудоемкости работ необходимо определить **график выполнения работ** и общие **сроки реализации проекта** — составить расписание работ по проекту.
- **Базовое расписание** — утвержденный план-график с указанными
 - временными фазами проекта,
 - контрольными точками
 - и элементами иерархической структуры работ.

Базовое расписание проекта

- Базовое расписание может быть наиболее наглядно представлено **диаграммой Ганта**.

- В этой диаграмме плановые **операции** или элементы иерархической структуры работ перечислены с левой стороны, **даты** отображаются сверху, а **длительность операций** показана горизонтальными полосками от даты начала до даты завершения.

Базовое расписание проекта



Базовое расписание проекта

- Базовое расписание это, как правило, **элемент контракта с заказчиком**.
- Контрольные точки (**вехи**) должны служить точками анализа состояния проекта и принятия решения «GO/NOT GO», поэтому они должны **зримо демонстрировать статус проекта**.
- Если работы **не связаны** между собой, то любую из них мы можем начинать и завершать, когда нам удобно.
- Все работы можно делать **параллельно** и в этом случае минимальная длительность проекта равна длительности самой долгой работы.

Базовое расписание проекта

- Однако, на практике между работами существуют **зависимости**, которые могут быть
 - «**жесткими**», например, анализ — проектирование — кодирование — тестирование и документирование конкретной функции;
 - или «**нежесткими**», которые могут пересматриваться или смягчаться.
- **Например**, последовательное выполнение задач конкретным исполнителем (можно перепланировать на другого исполнителя).
- В этом случае можно создавать «**заглушки**» эмулирующие работу базового ПО.

Базовое расписание проекта

■ Таким образом, диаграмма Ганта для расписания проекта выглядит как **гамак**, составленный из множества цепочек взаимосвязанных работ с **единой точкой начала и завершения**.

■ **Критический путь проекта** (Critical path) — самая длинная цепочка работ в проекте.

■ Увеличение длительности любой работы в этой цепочки приводит к увеличению длительности всего проекта.

■ В проекте всегда существует **хотя бы один критический путь**, но их может быть несколько.

Базовое расписание проекта

- Критический путь **может меняться** во время исполнения проекта.

- При исполнении проекта **руководитель** должен обращать внимание на исполнение задач на критическом пути в первую очередь и следить за появлением других критических путей.

- *Практическая рекомендация:*

- на критическом пути должны стоять работы с нежесткими связями, которые всегда можно перепланировать, если возникает угроза срыва сроков.

Базовое расписание проекта

- Чтобы проиллюстрировать понятие критического пути рассмотрим пример «**суперпроекта**».
- Концепция проекта выглядит следующим образом.
- **Цель проекта.** Сделать завтрак в постель
- **Результаты проекта.** Завтрак в постели из вареного яйца, тоста и апельсинового сока.
- **Ресурсы.** Имеется один оператор и обычное кухонное оборудование.
- **Сроки.** Проект начинается на кухне в 8:00 и завершается в спальне.

Базовое расписание проекта

- *Критерий приемки.* Используются минимальные трудовые ресурсы и срок. Конечный продукт имеет высокое качество: яйцо свежесваренное, тост теплый, сок холодный.
- *Обоснование полезности.* Проект служит достижению стратегических целей.
- *Иерархическая структура работ,* ориентированная на конечный продукт, с оценкой их длительности представлена на рис. 7.2.

Базовое расписание проекта

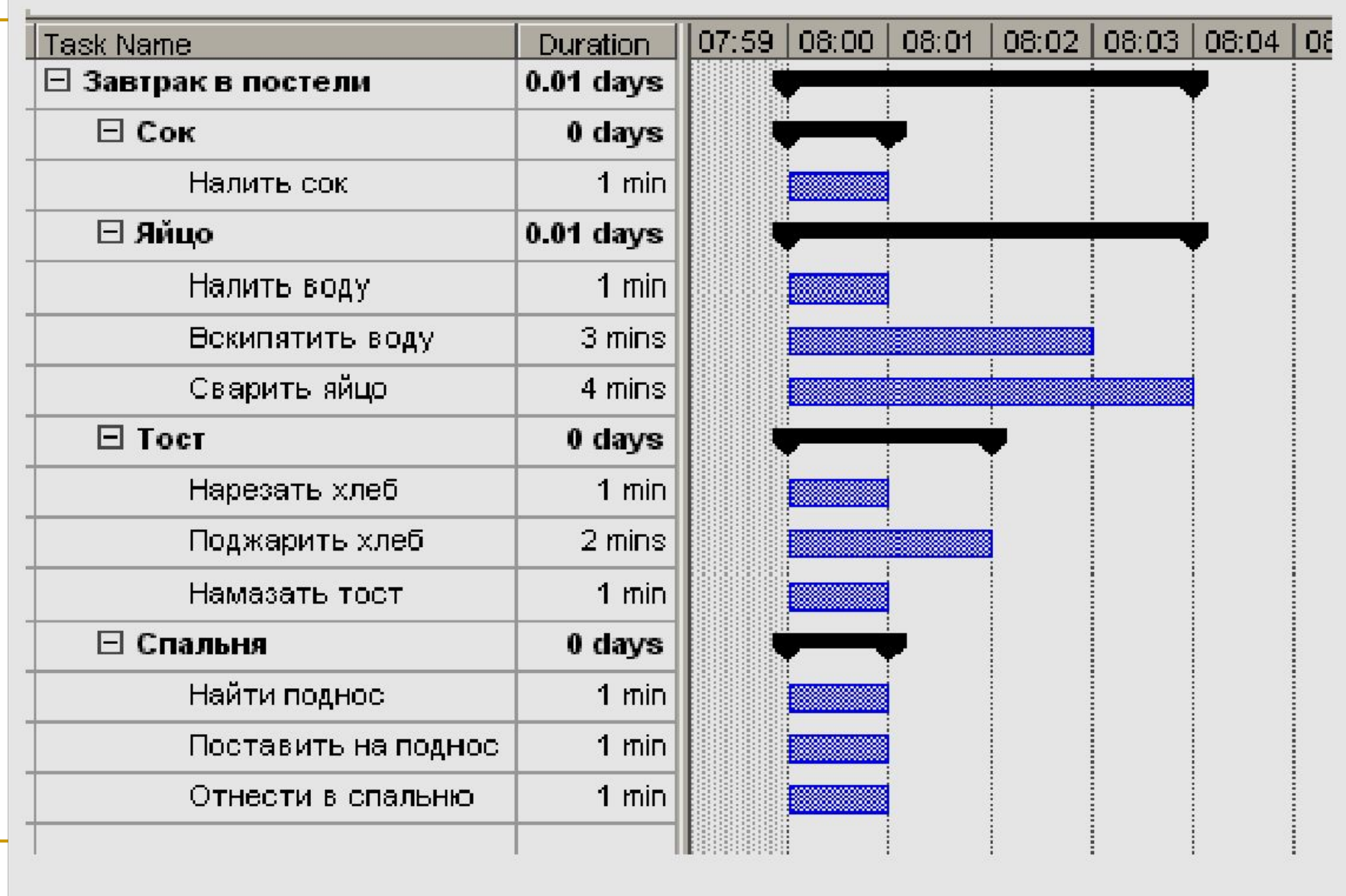


Рис. 7.2. Иерархическая структура работ «суперпроекта»

Базовое расписание проекта

- На следующем шаге мы должны учесть **зависимости между работами**, например, нельзя жарить хлеб, пока мы его не нарезали.
- С учетом зависимостей мы получим следующую диаграмму расписания нашего проекта (рис. 7.3).

Базовое расписание проекта

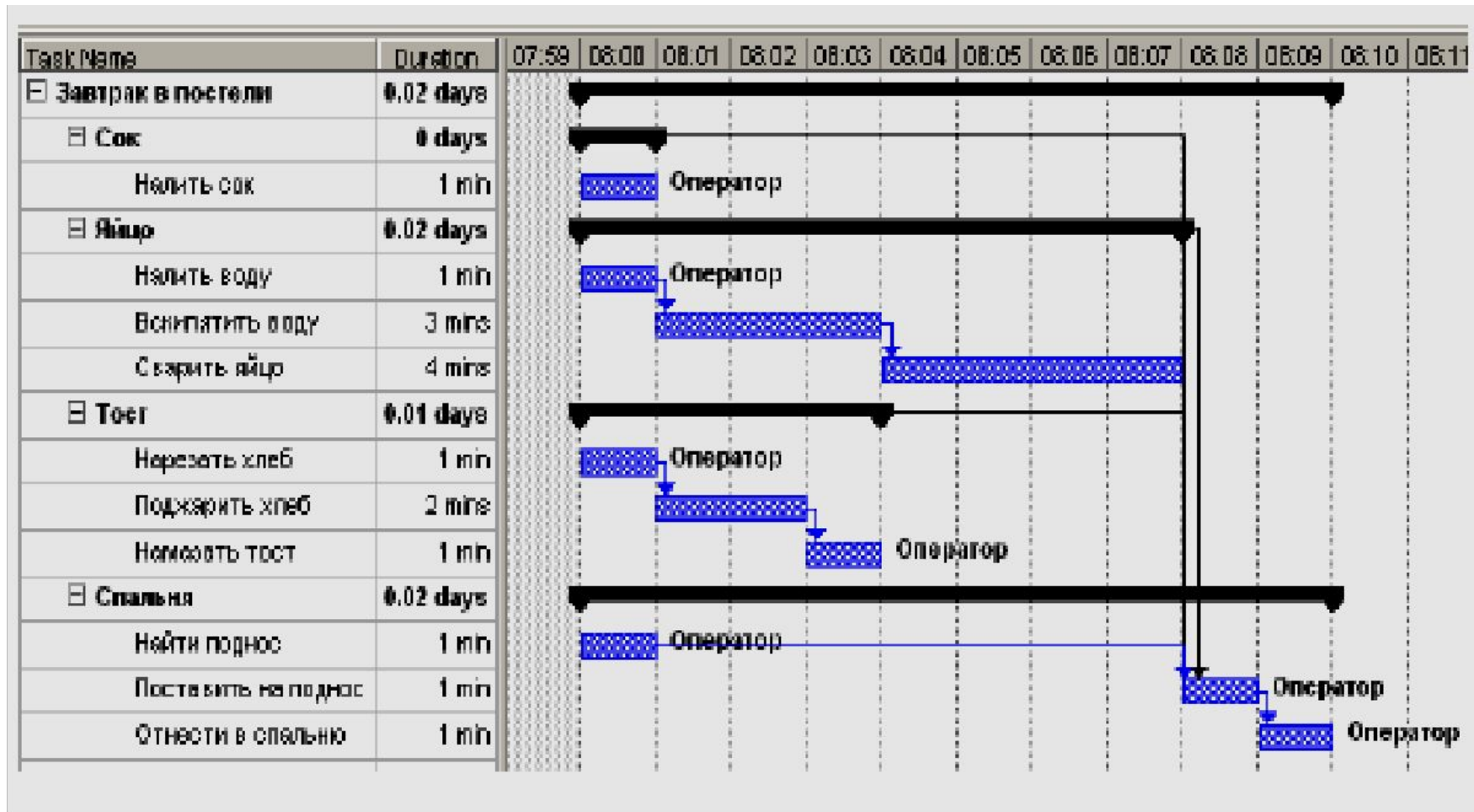


Рис. 7.3. Диаграмма расписания «суперпроекта» с учетом зависимостей между работами.

Базовое расписание проекта

- В результате мы определили, что **минимальный срок реализации** нашего проекта составляет **10 минут**.
- Однако мы не можем на этом остановиться, поскольку должны еще учесть **ограничение по ресурсам**.
- У нас только один оператор.
- Если мы посмотрим на диаграмму загруженности ресурсов (рис. 7.4.), то увидим, что наш **критический ресурс** загружен на первой минуте на 400% что недопустимо.

Базовое расписание проекта

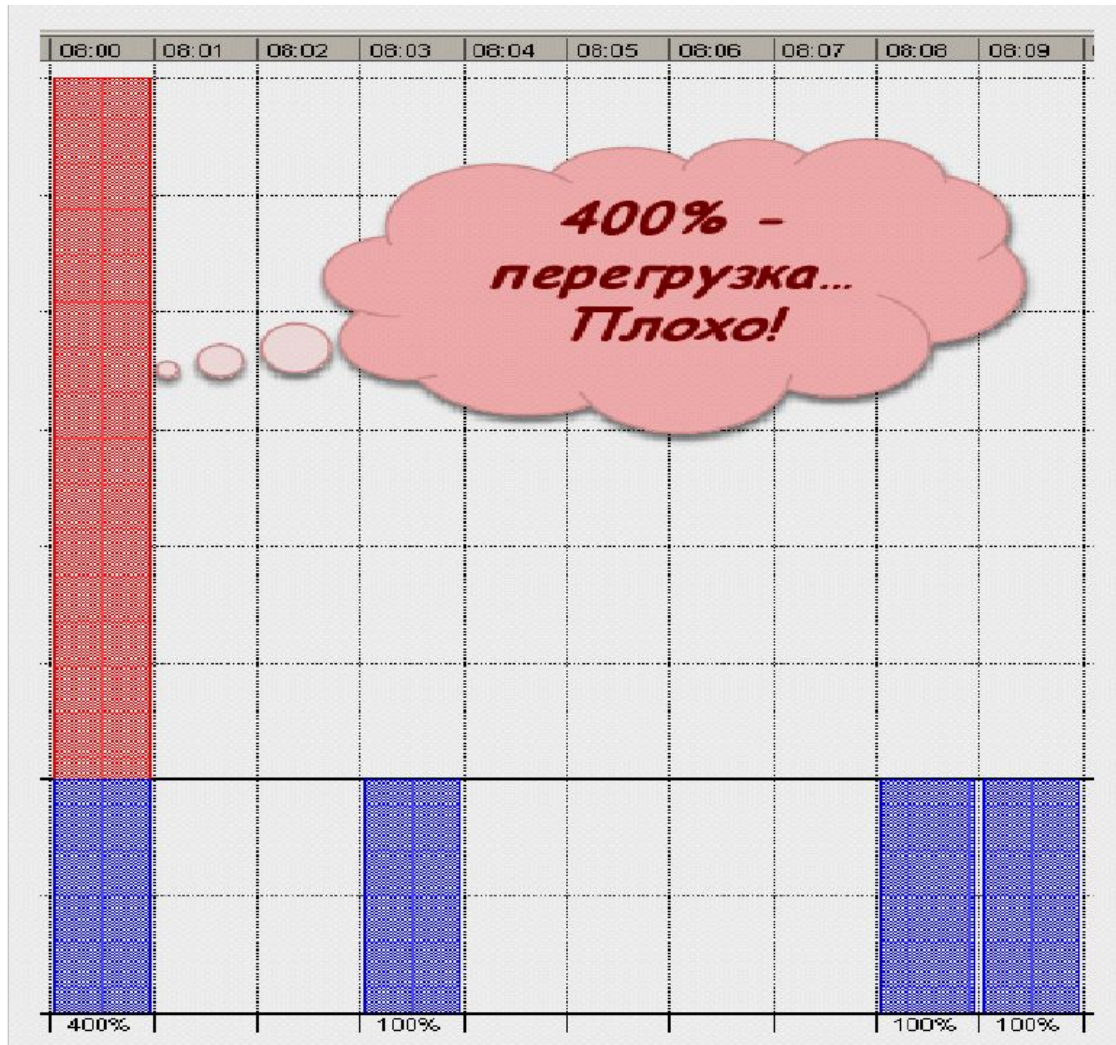


Рис. 7.4. Диаграмма загруженности ресурсов в «суперпроекте»

Базовое расписание проекта

- Следовательно, мы должны выполнить **выравнивание ресурсов**.
- Поскольку одним из критериев успеха проекта является его **минимальная длительность**, то если мы не хотим ее увеличивать, мы должны выявить критический путь в проекте (рис. 7.5) и не сдвигать работы, которые на нем находятся.

Базовое расписание проекта

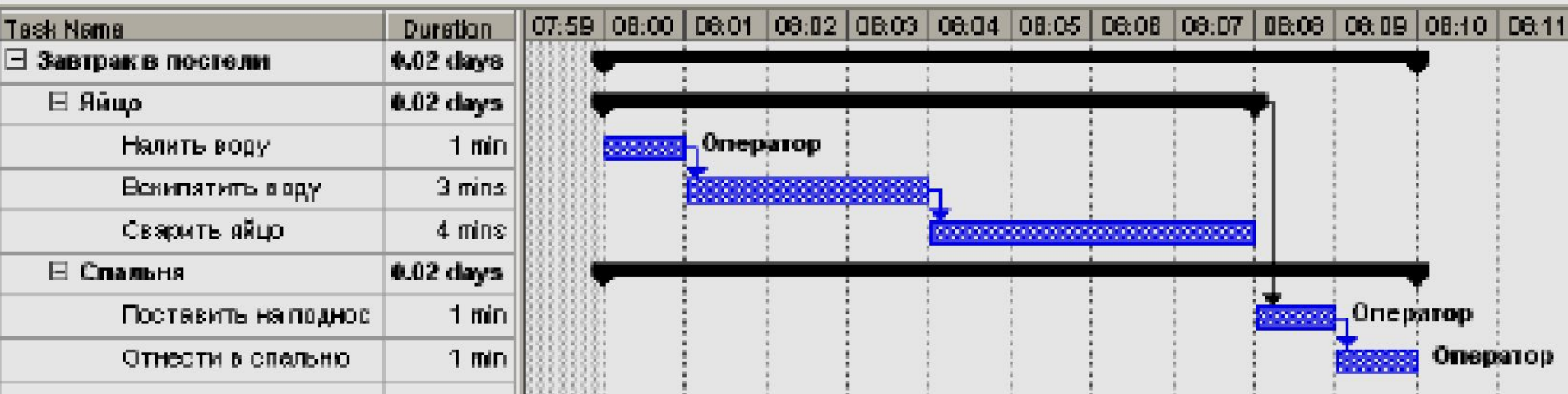


Рис. 7.5. Критический путь в «суперпроекте»

Базовое расписание проекта

Поэтому, после выравнивания ресурсов, расписание нашего проекта будет выглядеть следующим образом (рис. 7.6).

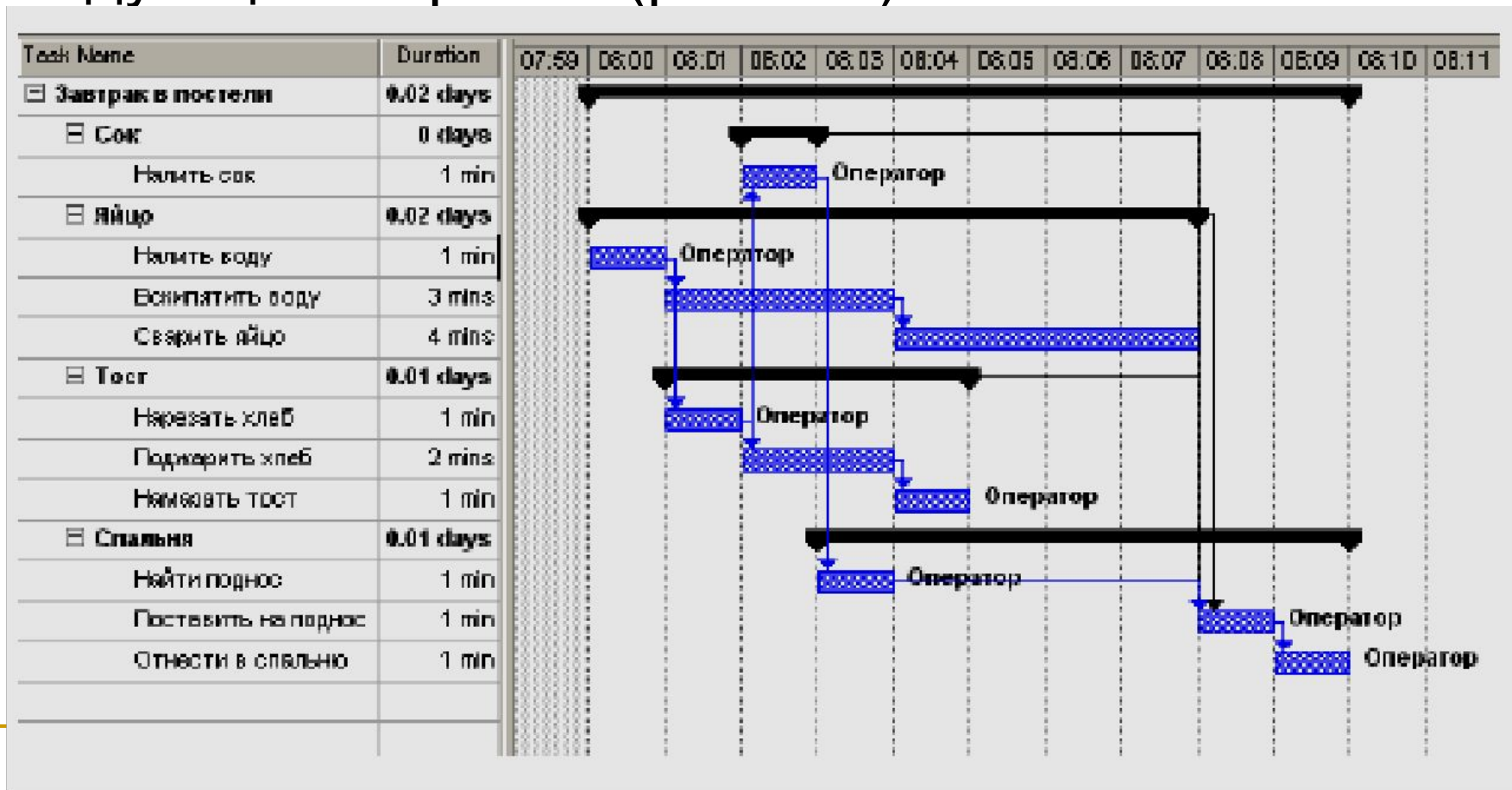


Рис. 7.6. Расписание «суперпроекта» после выравнивания ресурсов

Базовое расписание проекта

- Теперь **диаграмма загруженности ресурсов** (рис. 7.7) выглядит приемлемо и у оператора даже появилось три минуты свободного времени на перекур.
- При этом общая длительность реализации проекта по-прежнему составляет 10 минут.

Базовое расписание проекта

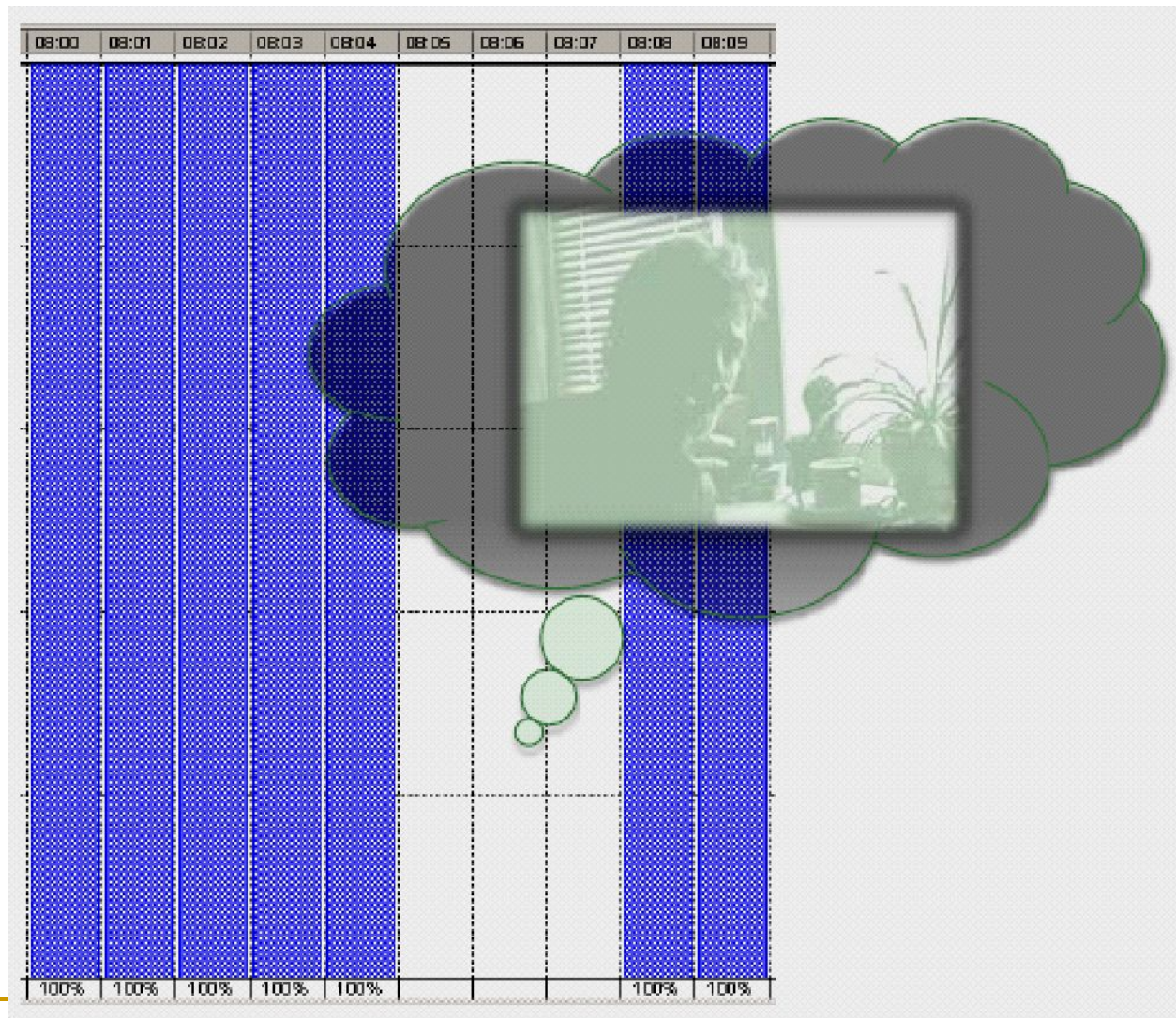


Рис. 7.7. Диаграмма загрузки ресурсов после выравнивания

Выводы

- На верхнем уровне **Иерархической Структуры Работ (ИСР)** проекта должны находиться не процессы, а **продукты проекта**, на следующем уровне — компоненты из которых эти продукты состоят.

- **Выделение компонентов**, составляющих программный продукт,

- это **элемент высокоуровневого проектирования**,

- которое мы должны выполнить на фазе планирования проекта,

- ~~не дожидаясь проработки всех функциональных требований к разрабатываемому ПО.~~

Выводы

■ Помимо работ, непосредственно направленных на создание программного обеспечения, в плане проекта должны быть предусмотрены **необходимые ресурсы для обеспечения работ по следующим процессам:**

- управление **содержанием**;
- управление **конфигурациями**,
- управление **качеством**,
- управление **рисками**,
- управление **проектом**.

Выводы

- В проекте всегда существует **хотя бы один критический путь**, но их может быть несколько.
- **Критический путь может меняться во время** исполнения проекта.
- При исполнении проекта **руководитель** должен обращать внимание **на исполнение задач на критическом пути** в первую очередь и следить за **появлением других критических путей**.

Вопросы

- 1. Что такое Иерархическая структура работ (ИСР)?
- 2. Что такое *инкрементальный* процесс разработки коммерческого ПО?
- 3. Что включает в себя план управления содержанием проекта?
- 4. Что включает в себя планирование организационной структуры?
- 5. Что включает в себя планирование управления конфигурациями?
- 6. Какие работы включает в себя план управления качеством?
- 7. Что такое базовое расписание проекта?
- 8. Нарисуйте пример диаграммы Ганта.
- 9. Что такое критический путь проекта?
- 10. Как и по каким критериям может осуществляться выравнивание ресурсов в базовом расписании проекта?

Лекция

ПЛАНИРОВАНИЕ ПРОЕКТА

Король Иван Андреевич
