

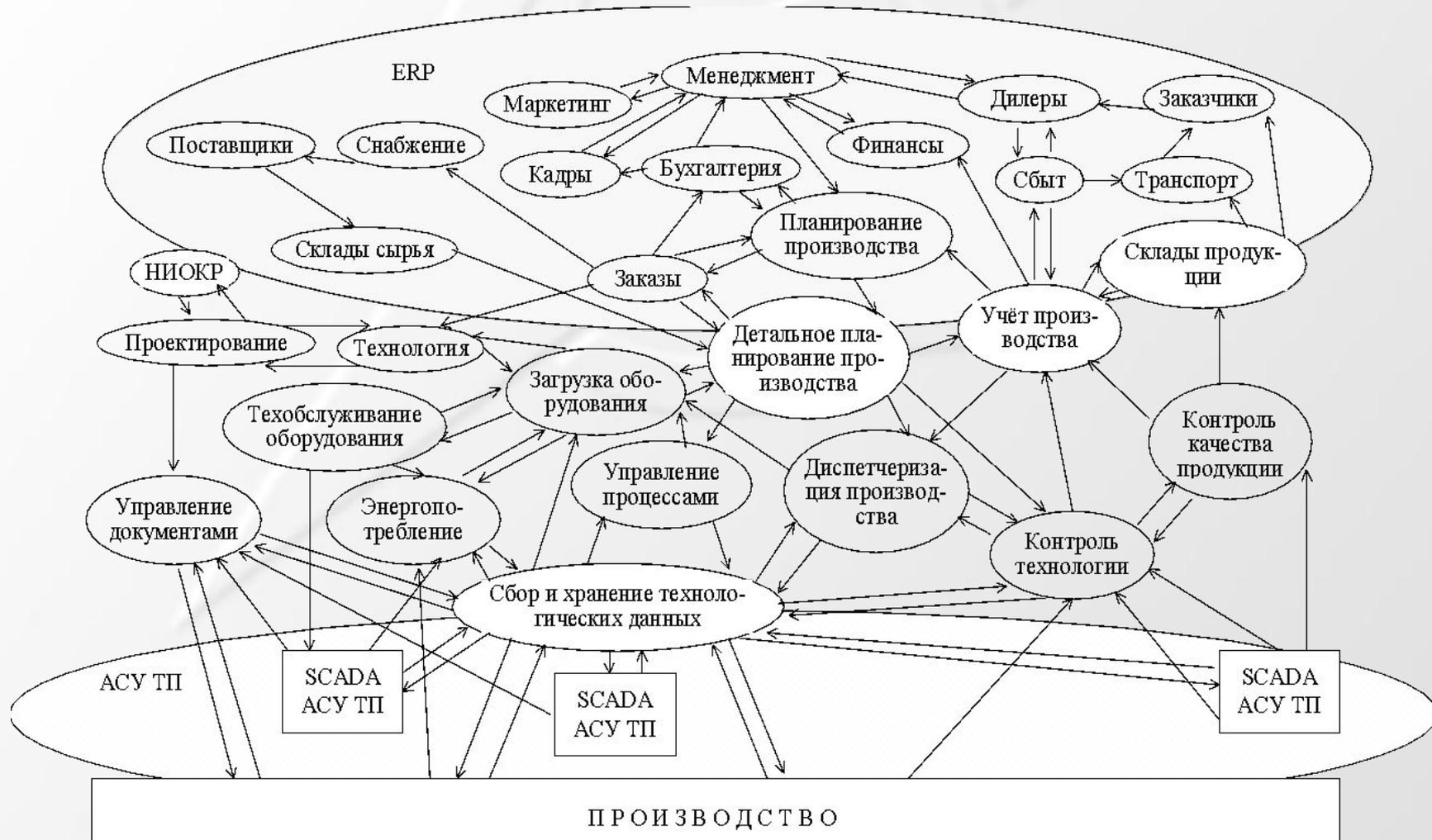
Методология программной инженерии (спецификация требований)

Лекция 1 Жизненный цикл программного обеспечения и его модели.

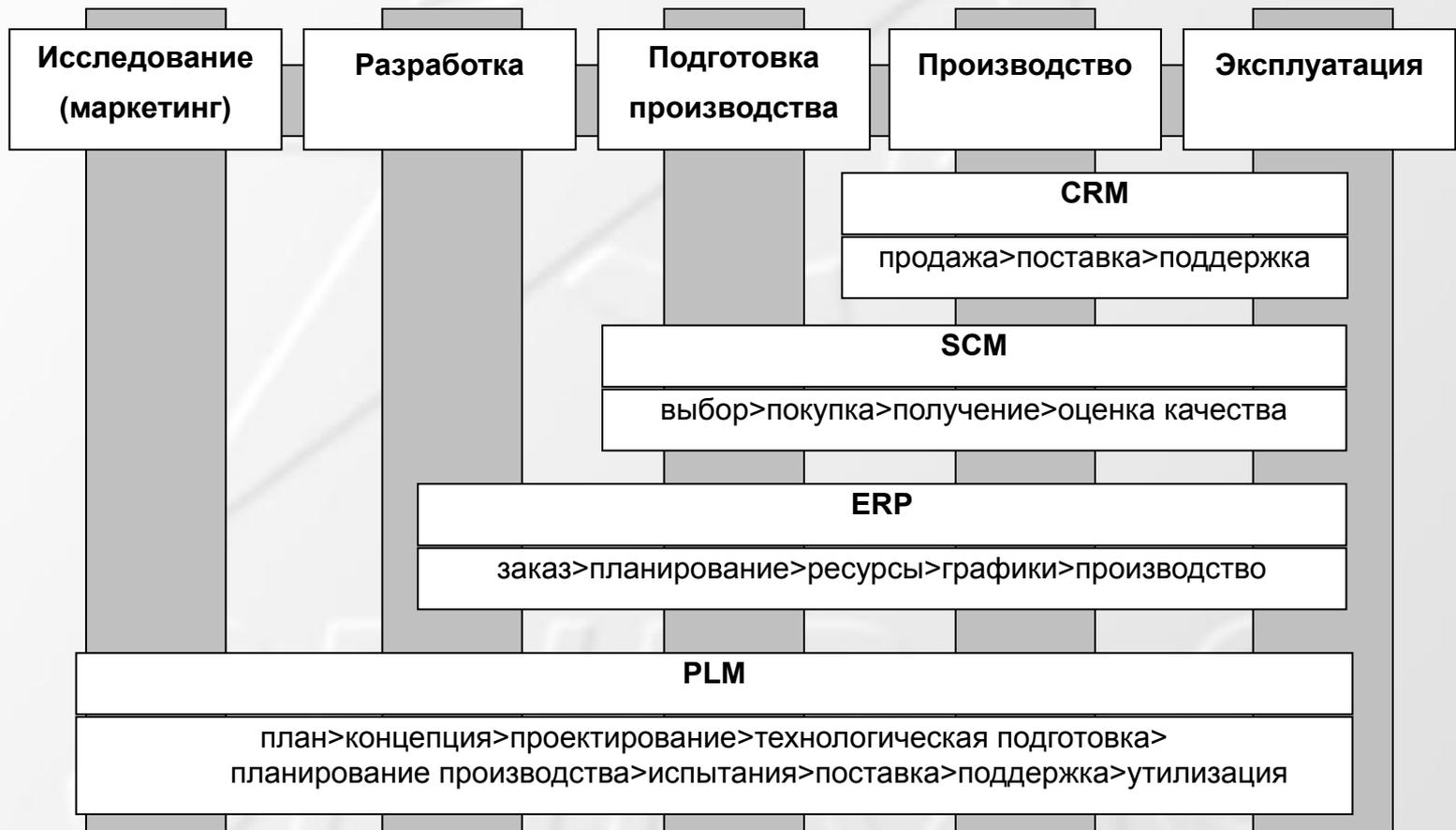
Доктор технических наук, профессор Соколов Б.В

***С.-Петербургский институт информатики и автоматизации РАН,
С.-Петербург,
14 линия ВО, 39, СПИИ РАН,***

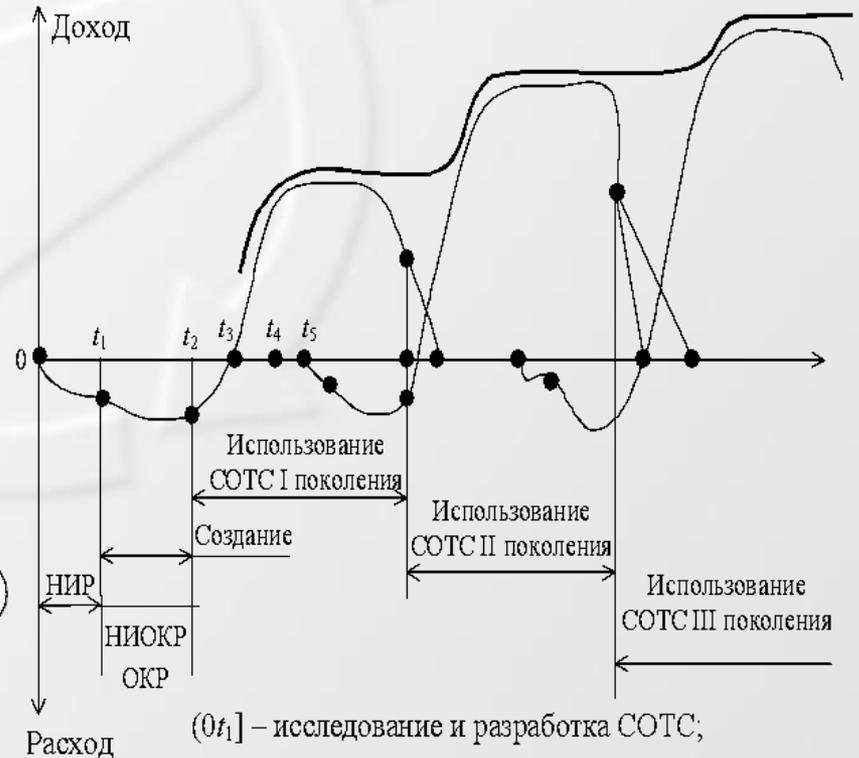
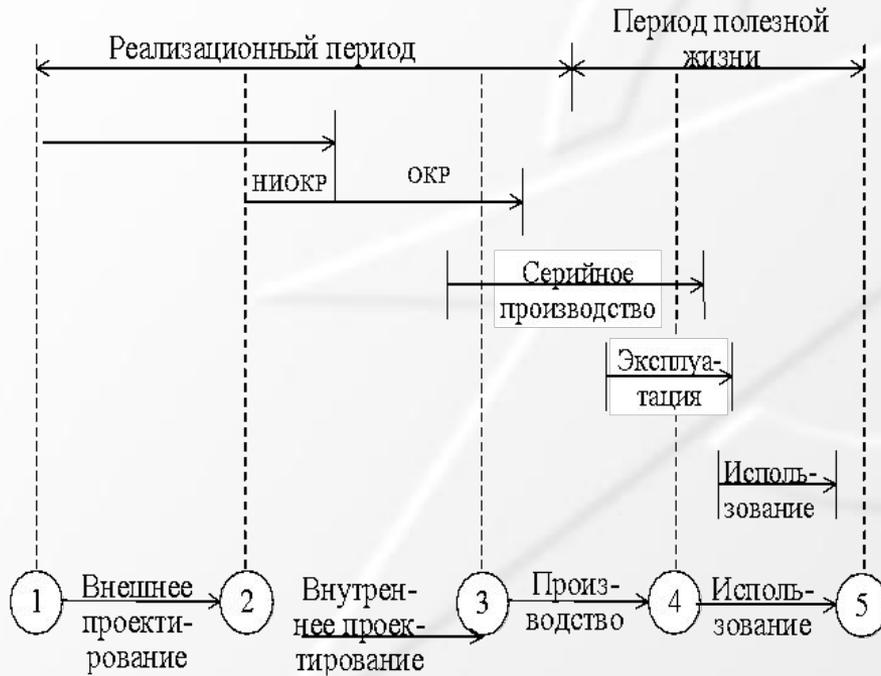
Модели ЖЦ ПО



Модели ЖЦ ПО



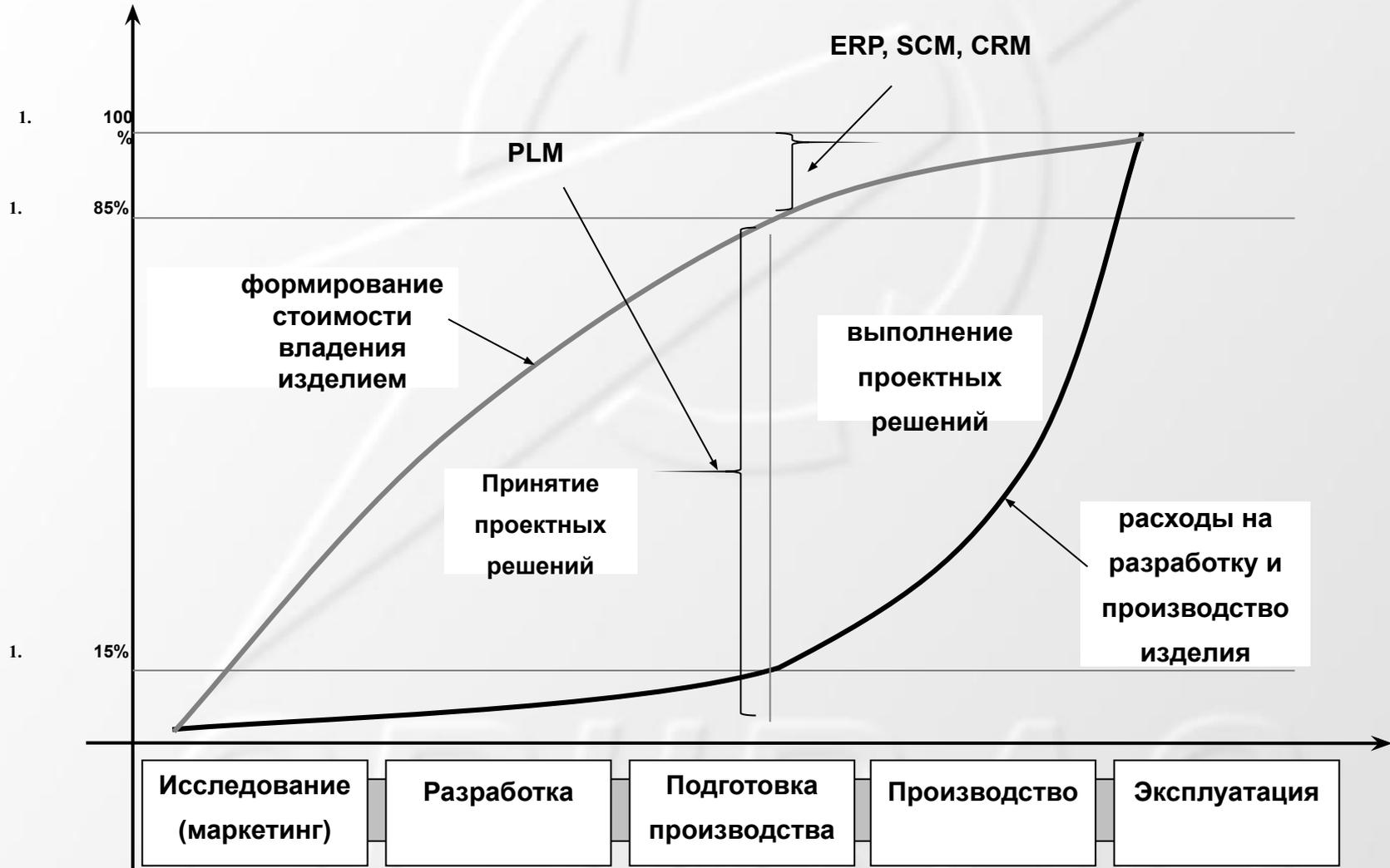
Модели ЖЦ ПО



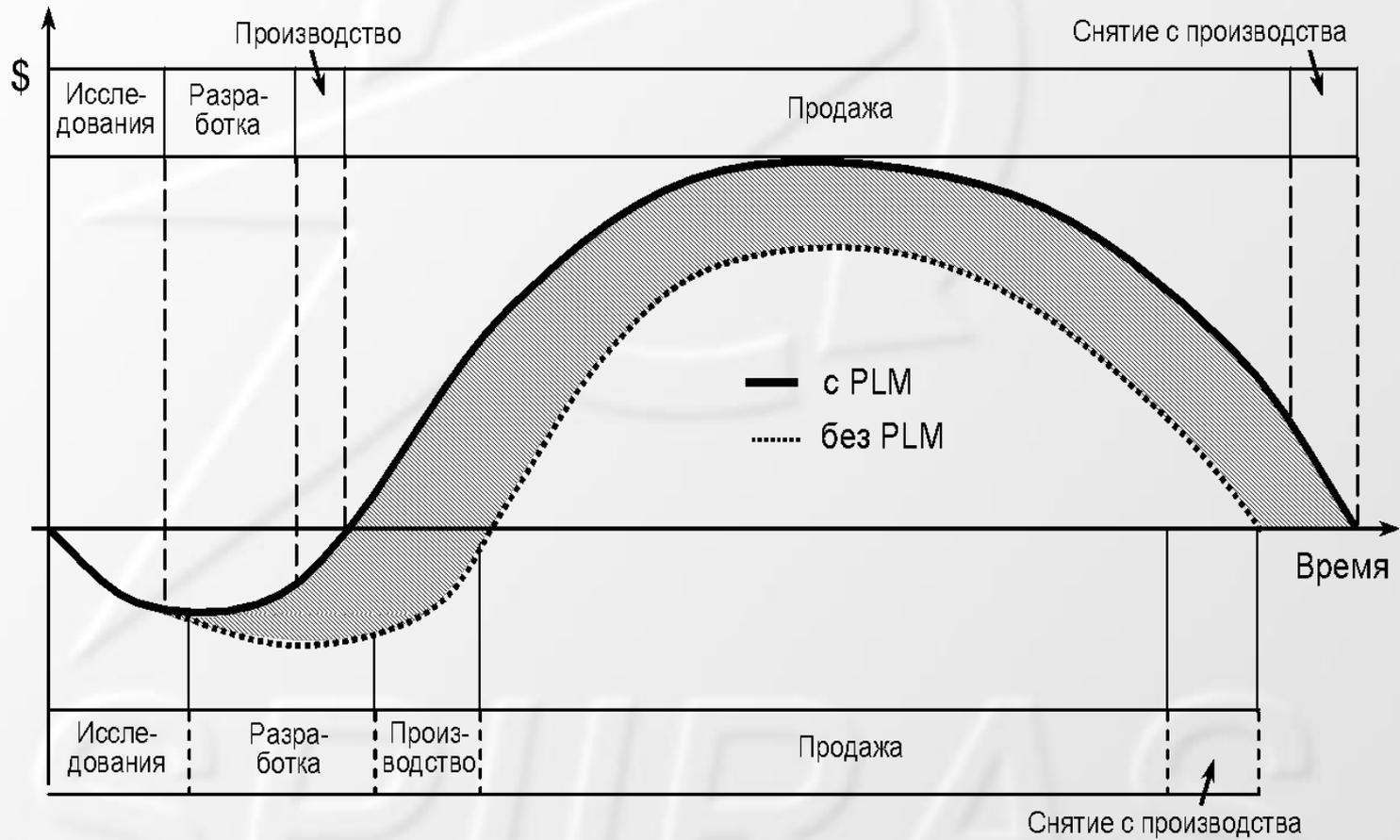
- ① – принятие решения о разработке СОТС
- ② – утверждение технического задания СОТС
- ③ – принятие решения о серийном производстве СОТС
- ④ – ввод в эксплуатацию СОТС
- ⑤ – решение о модернизации и развитии СОТС

- $(0t_1]$ – исследование и разработка СОТС;
- $(t_1t_2]$ – освоение и организация производства элементов и подсистем СОТС
- $(t_2t_3]$ – время окупаемости СОТС
- $(t_3t_4]$ – интервал времени, соответствующий сроку второй наработки на вложения

Модели ЖЦ ПО

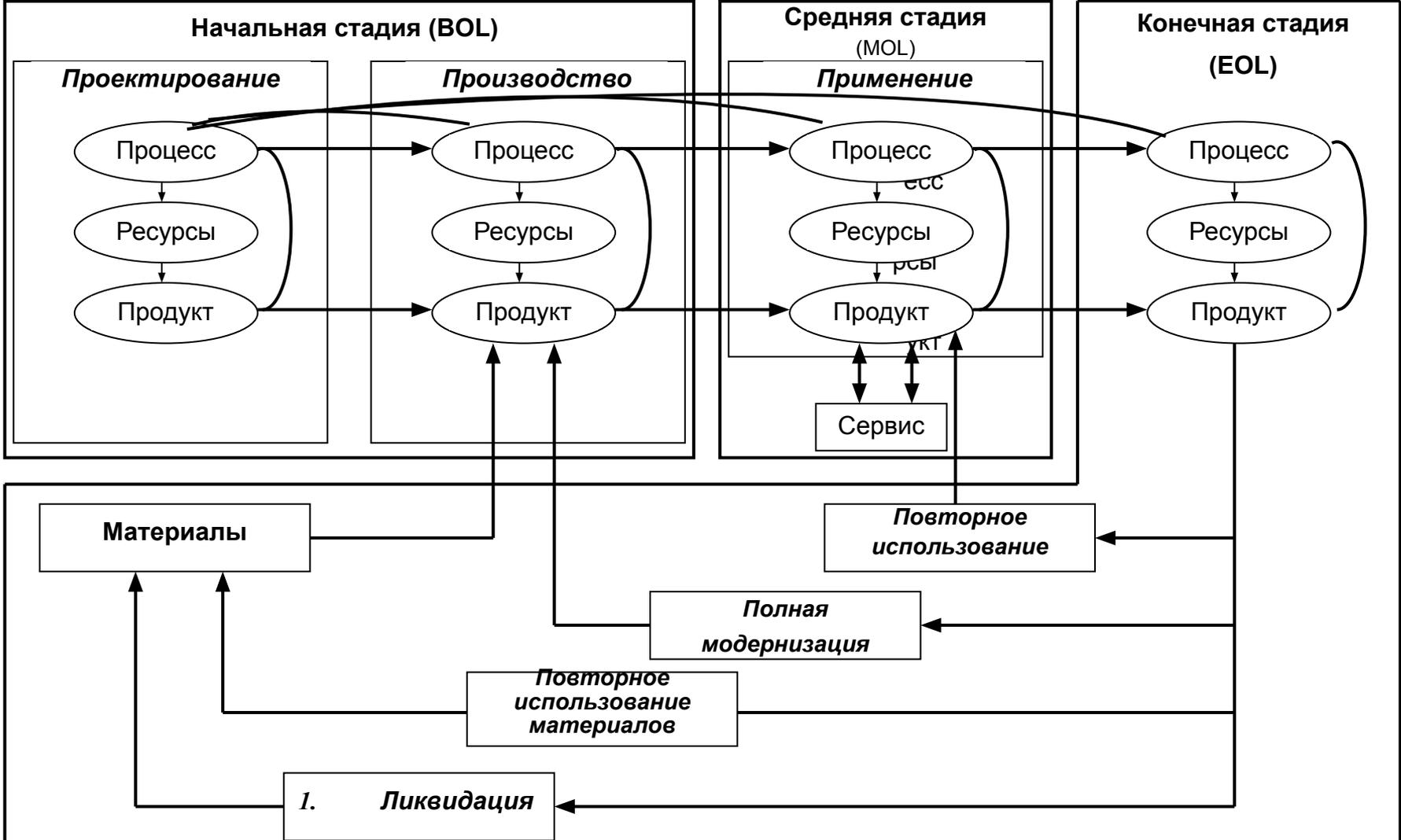


Модели ЖЦ ПО



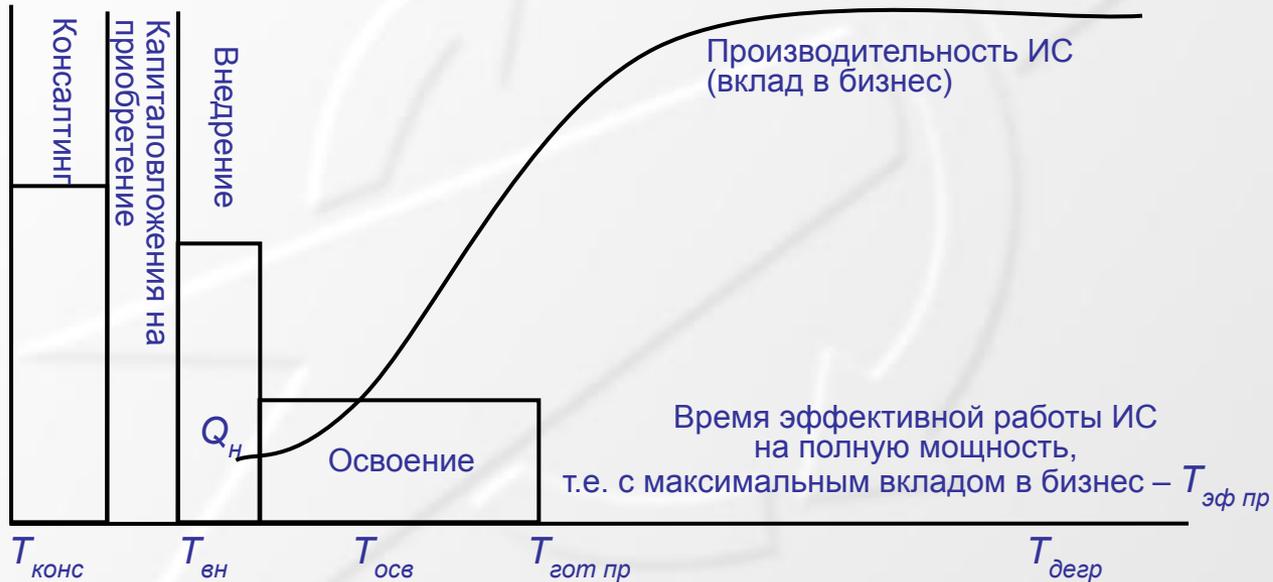
Модели ЖЦ ПО

1. ЖИЗНЕННЫЙ ЦИКЛ ПРОДУКЦИИ



Модели ЖЦ ПО

Издержки



Амортизация

Гарантийный срок $T_{гар}$

Сопровождение, поддержка

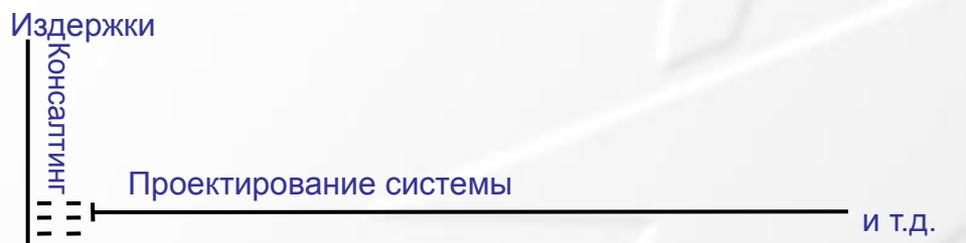
Испытания

Сначала испытаний бывает много, затем их объем может уменьшаться

Обслуживание

На этапе приработки интенсивность и объем обслуживания бывают больше

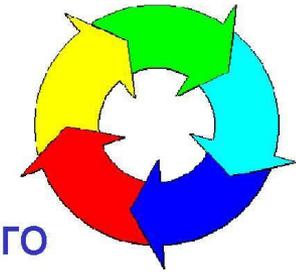
Модели ЖЦ ПО



Модели ЖЦ ПО



Значимость модели ЖЦ



- Это каркас для определения повторяемого процесса, в явном виде задающего деятельности по созданию высококачественного ПО
- Она дает разработчику возможность включать лучшие практики из опыта других и делиться своими лучшими практиками с другими
- Понятие модели ЖЦ применимо к программным проектам любого размера, большим и малым

Типичные фазы проекта в моделях ЖЦ

- на шаге анализа:

- планирование проекта – project planning
- сбор и анализ требований – requirements



- на шаге проектирования:

- предварительное (высокоуровневое) проектирование – preliminary (high-level) design
- подробное (детальное) проектирование – detailed design



- на шаге реализации:

- кодирование – coding



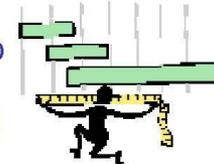
- на шаге обзора

- тестирование – testing



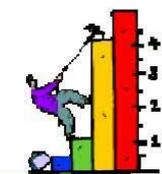
Полезные следствия принятия модели ЖЦ

- Модель способствует определению требований до проектирования системы — надо определить, что система должна делать **ДО** ее создания
- Модель способствует проектированию ПО до построения его компонентов — надо спланировать, как именно компоненты будут взаимодействовать между собой **ДО** их создания
- Модель определяет, какие именно рабочие продукты должен поставить данный процесс разработки — надо сгенерировать стандартный набор поставок, которые тестируемы и могут помочь в сопровождении системы
- Модель дает возможность руководству проектом тщательно отслеживать его продвижение — руководство должно располагать базовыми стандартами для измерения качества продукта и производительности как самого процесса разработки, так и разработчиков
- Модель снижает затраты на разработку и сопровождение — все предыдущее этому способствует
- Модель дает возможность организации-разработчику быть более структурированной и управляемой



Преимущества повторяемого процесса

- Улучшает качество продукта через согласованность в разработке — продукт определяется как состоящий из всех поставляемых рабочих продуктов его жизненного цикла
- Облегчает управление проектами — сравнение со стандартами выявляет проблемы, требующие решения
- Облегчает отслеживание состояния проекта — определение деятельности и заданий в процессе является средством знать, что именно было сделано, за какое время и какими ресурсами
- Дает базу для улучшений и измерений — успех проекта измеряется сравнением завершенных компонентов со стандартами и фактической производительности с ожидаемой по оценкам. Производительность ниже базовой указывает на необходимость улучшений в процессе. Качество продукта ниже базового указывает на необходимость исправлений в продукте
- Исправляя организационные слабости, способствует повышению уровня зрелости — когда недостатки процесса исправляются, организация переходит на более высокий уровень зрелости, как это определяется моделями зрелости CMM/CMMI

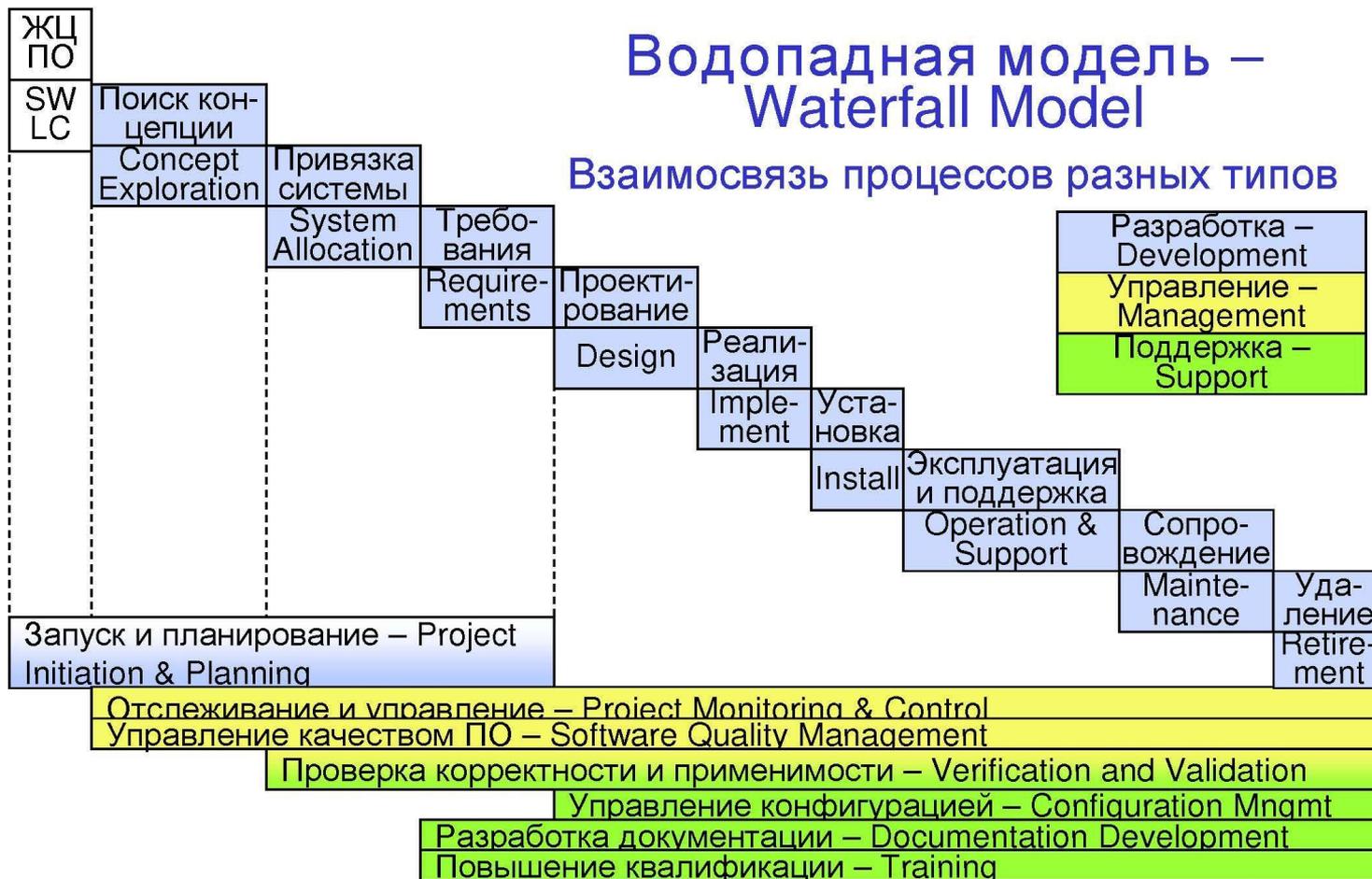


Фазы простого процесса для малых проектов

- **Спланировать – Plan**
 - Определить функциональные требования, привязав их к ПО и к аппаратуре
 - Изготовить план разработки, включая определение поставляемых рабочих продуктов, оценки ресурсов, определение процессов поддержки
- **Специфицировать – Specify**
 - Точно определить внешние характеристики будущей программы с точки зрения пользователя (напр., через прототипирование и предъявление пользователям)
 - Провести обзор спецификаций как минимум с участием инженера-проектировщика, основных пользователей и заказчика
- **Разработать – Develop**
 - Выполнить детальный проект с точки зрения его реализации
 - Провести обзор детального проекта с участием разработчиков и тестировщиков
 - Выполнить кодирование этого проекта
 - Провести обзор кода с участием тех же лиц, что и в обзоре проекта
 - Скомпилировать программу, выявив синтакс. ошибки, не вскрытые на обзоре кода
 - Протестировать программу, выявив логич. ошибки, не вскрытые обзором кода
- **После завершения – Postmortem**
 - Проанализировать проделанную работу, проверив что все плановые поставки завершены, сравнив результаты работ с требованиями пользователя по качеству
 - Сравнить фактические результаты с первоначальными оценками, объяснив все расхождения

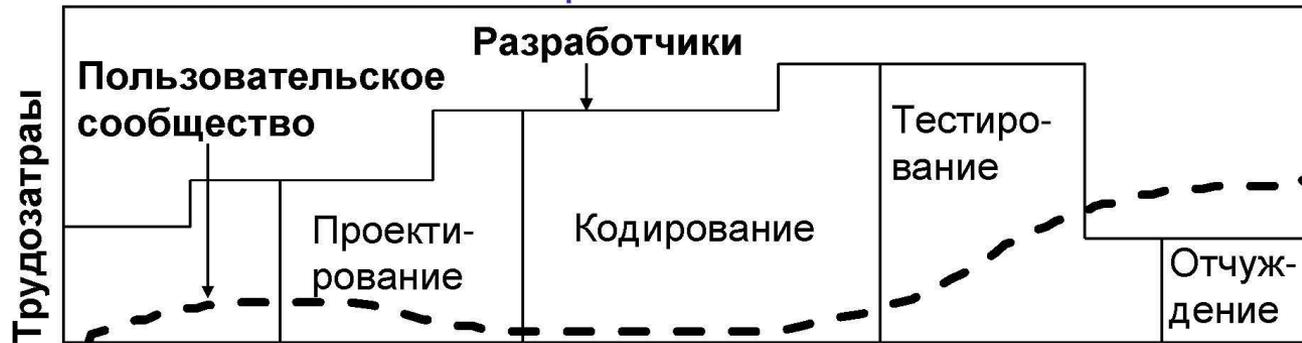
Водопадная модель – Waterfall Model

Взаимосвязь процессов разных типов

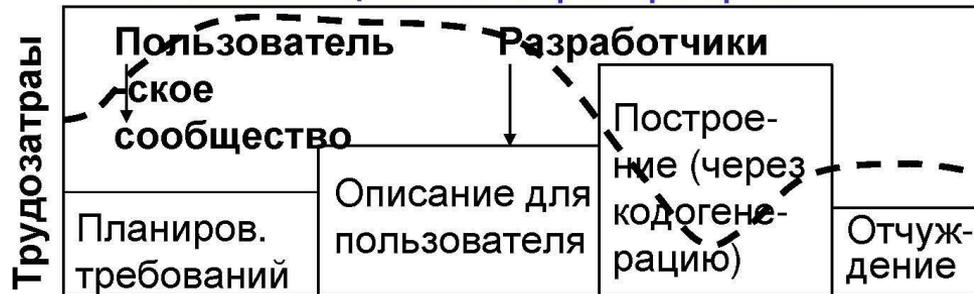


Модель быстрой разработки приложений - Rapid Application Development Model (RAD)

Обычный жизненный цикл



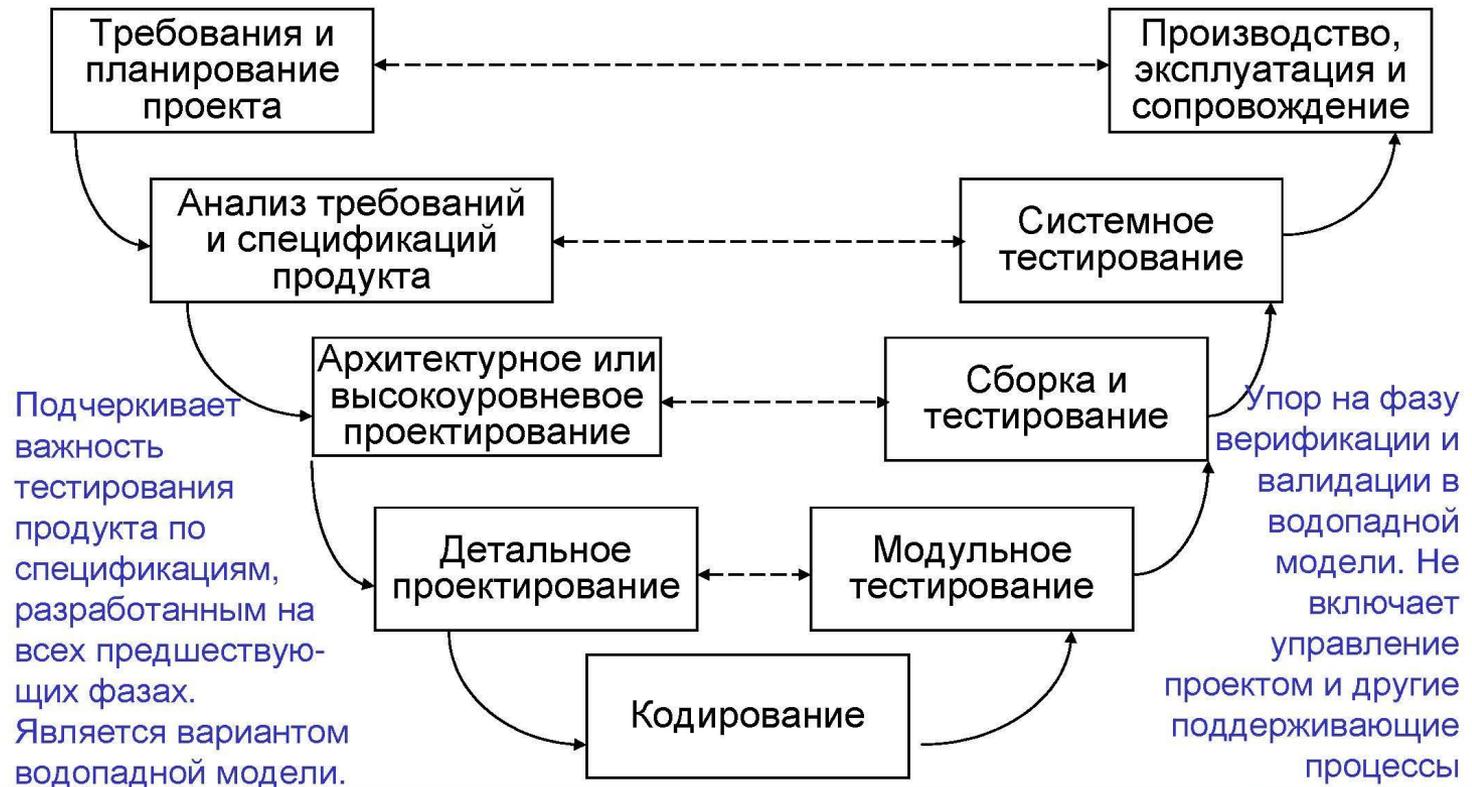
Жизненный цикл быстрой разработки



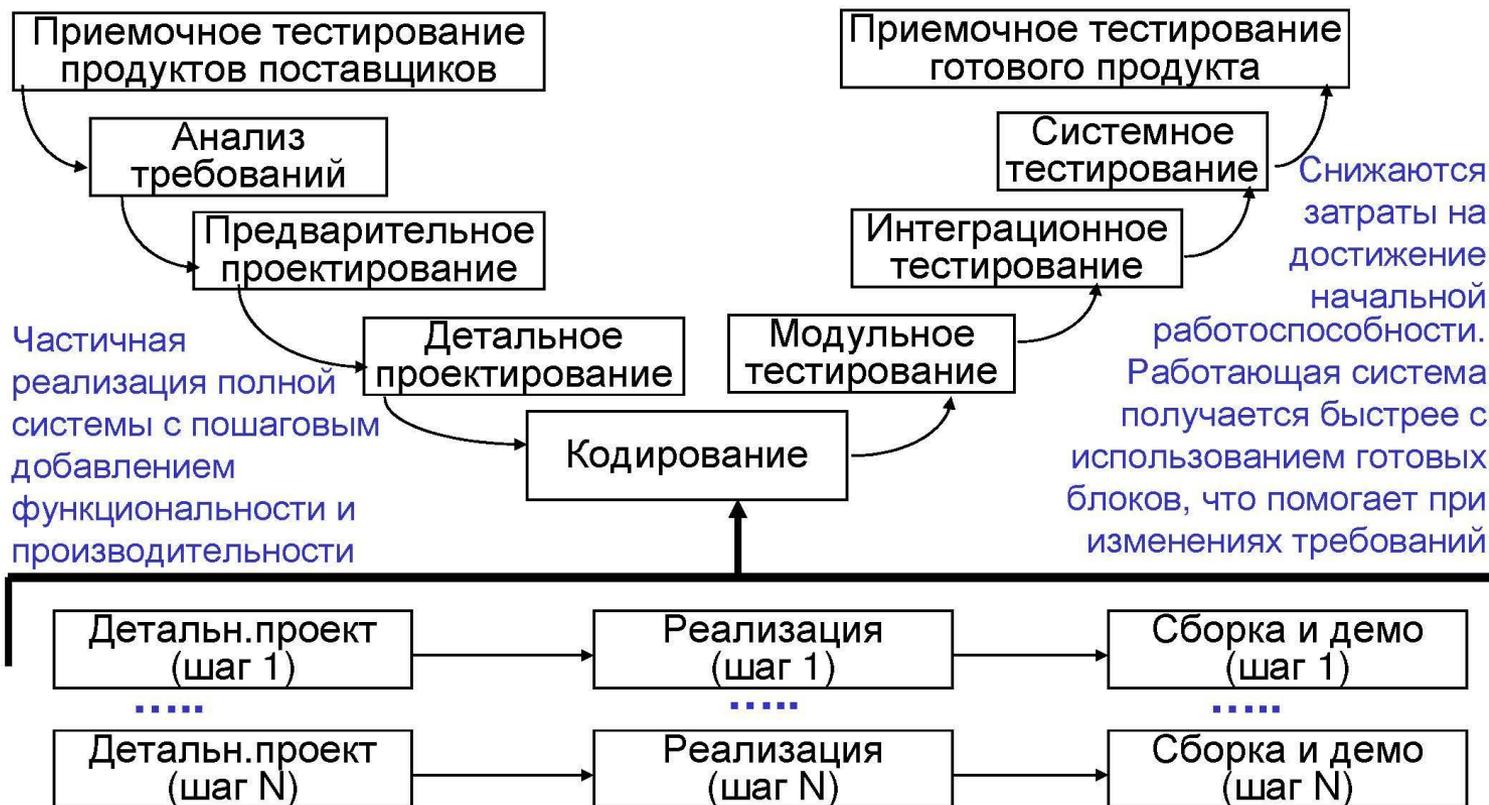
Время →

Роль пользователей критична. В обычном ЖЦ большая часть работы – программирование и тестирование. В БРП за счет кодогенерации большая часть работы – планирование и проектирование.

V-образная модель - V-shaped Model



Пошаговая модель - Incremental Model



Спиральная модель Боэма - Boehm's Spiral Model

Определить цели, альтернативы, ограничения

Суммарная стоимость ↑

Оценить альтернативы, выявить и решить риски



Факторы, влияющие на выбор модели ЖЦ - 1

1. **Доступность ресурсов:** *низкая* или *некоторая* – ресурсы нельзя оценить или они недоступны; *высокая* – ресурсы можно определить и они доступны
2. **Сложность проекта:** *низкая* – все критерии сложности низкие; *средняя* – критерии сложности смешаны: 1-2 высокие, 1-2 низкие; *высокая* – все критерии сложности высокие
3. **Стоимость приложения:** *низкая* – оценка меньше некоторой суммы; *средняя* – оценка равна этой сумме; *высокая* – оценка больше этой суммы
4. **Стоимость будущих обновлений:** *низкая* – оценка меньше некоторой суммы; *высокая* – оценка стоимости больше этой суммы
5. **Дискретное изменение требований:** *малое* – затрагивает не более 5 интерфейсов и включает не более 10 процессов; *большое* – затрагивает более 5 интерфейсов и включает более 10 процессов
6. **Легкость в использовании:** *просто* – фазы и поставки понятны, процесс сфокусирован на разработку, а не на поддерживающие процессы; *сложно* – поддерживающие процессы требуют управления, наряду с разработкой
7. **Функциональные потребности приложения:** *смутные* – трудно определяемые; разработчики формулируют их сами; *специфицированные* – хорошо определены, если они измеряемы и тестируемы
8. **Постепенное изменение требований:** *малое* – затрагивают небольшой объем системы; *большое* – требуют пересмотра большого числа интерфейсов
9. **Время жизни приложения:** *краткое* – краткосрочное решение, например, на несколько дней; *среднее* – от 3 до 5 лет; *долгое* – более 5 лет

Факторы, влияющие на выбор модели ЖЦ - 2

10. **Технология производства продукта:** *существующая* – современная технология, уже применявшаяся разработчиками; *новая* – технология, ранее не применявшаяся данными разработчиками
11. **Отдача приложения:** *низкая* – результаты анализа стоимость-выгоды меньше заданного значения или если выгоды несущественны; *высокая* – результаты анализа стоимость-выгоды больше заданного значения
12. **Качество результатов:** *сразу* – нужное качество достигается с первого раза; *переработка* – для достижения нужного качества требуются переделки
13. **Изменчивость требований:** *низкая* – требования изначально хорошо заданы и стабильны; *средняя* – минимальные изменения в требованиях допустимы; *высокая* – при высокой изменчивости эффект движущейся мишени
14. **Повторное использование продуктов и документации:** *низкое* – возможность использования продуктов из других реализаций ограничена; *высокое* – предполагается их использование в интеграции и тестировании
15. **Перспективы управления рисками:** *нет* – не рассматривается; *да* – управление рисками должно быть включено в данный ЖЦ
16. **Неопределенность требований:** *нет* – требования определены и предсказуемы; *да* – существует неопределенность, ЖЦ должен это учитывать
17. **Неизвестные требования:** *нет* – все требования выявлены; *да* – некоторые требования могут быть упущены

Сравнительные характеристики 6 моделей ЖЦ

Модель	Характеристики
Водопадная	<ul style="list-style-type: none"> • Проста и очевидна в использовании • Обеспечивается необходимый строгий контроль за разработкой • Программный продукт – только по завершении всего цикла разработки • Изменения в требованиях не предполагаются или ограничены
Быстрой разработки приложений	<ul style="list-style-type: none"> • Малые команды опытных и мотивированных разработчиков • Сокращение цикла разработки и рост производительности труда • Повторное использование через автоматическую кодогенерацию
V-образная	<ul style="list-style-type: none"> • Проста в использовании • Упор на соотнесение фаз тестирования и фаз разработки
Пошаговая	<ul style="list-style-type: none"> • Дает работающую систему быстрее • Снижает вероятность изменения требований пользователя во время разработки • Устраняет необходимость перехода от настоящего к будущему за 1 шаг
Спиральная	<ul style="list-style-type: none"> • Включает в себя водопадную модель • Разбивает фазы разработки на более мелкие шаги • Допускает гибкость в проектировании • Движущей силой является анализ рисков • Пользователи видят работающий продукт раньше
Прототипная	<ul style="list-style-type: none"> • Создает «быструю» частичную реализацию до прояснения требований • Обратная связь от пользователей по сильным и слабым сторонам • Изначально требования не ясны

Матрица выбора модели ЖЦ



Критерии выбора	V-образн.	Водопадная	Прототипная	Пошаговая	Спиральная	БРП
1. Доступность ресурсов	низкая	низкая	некотор.	некотор.	некотор.	некотор.
2. Сложность проекта	низкая	низкая	средняя	высокая	высокая	средняя
3. Стоимость приложения	низкая	низкая	низкая	средняя	высокая	низкая
4. Стоимость будущих обновлений	высокая	высокая	низкая	низкая	низкая	низкая
5. Дискретное изменение требований	большое	большое	малое	малое	малое	малое
6. Легкость в использовании	просто	просто	просто	сложно	сложно	просто
7. Функциональные потребности приложения	специфич.	специфич.	смутные	смутные	смутные	специфич.
8. Постепенное изменение требований	малое	малое	малое	большое	большое	малое
9. Время жизни приложения	среднее	среднее	краткое	долгое	долгое	краткое
10. Технология производства продукта	существ.	существ.	новая	новая	новая	новая
11. Отдача приложения	высокая	высокая	низкая	высокая	высокая	низкая
12. Качество результатов	перераб.	перераб.	сразу	сразу	сразу	перераб.
13. Изменчивость требований	низкая	низкая	низкая	средняя	высокая	низкая
14. Повторное использование продукта и технологий	низкое	низкое	низкое	высокое	высокое	низкая
15. Перспективы управления рисками	нет	нет	да	нет	да	нет
16. Неопределенность требований	нет	нет	да	да	да	нет
17. Неизвестные требования	нет	нет	да	да	да	нет

Комбинированные модели ЖЦ

Пример Microsoft DOS 6.0

Компонент	Модель жизненного цикла его разработки
Virus Backup	Внешняя разработка с последующей интеграцией
Fixes	V-образная
Disk Doublespace	Водопадная
File Recovery	Спиральная
Memory Management	Прототипная
Весь релиз 6.0	Пошаговая

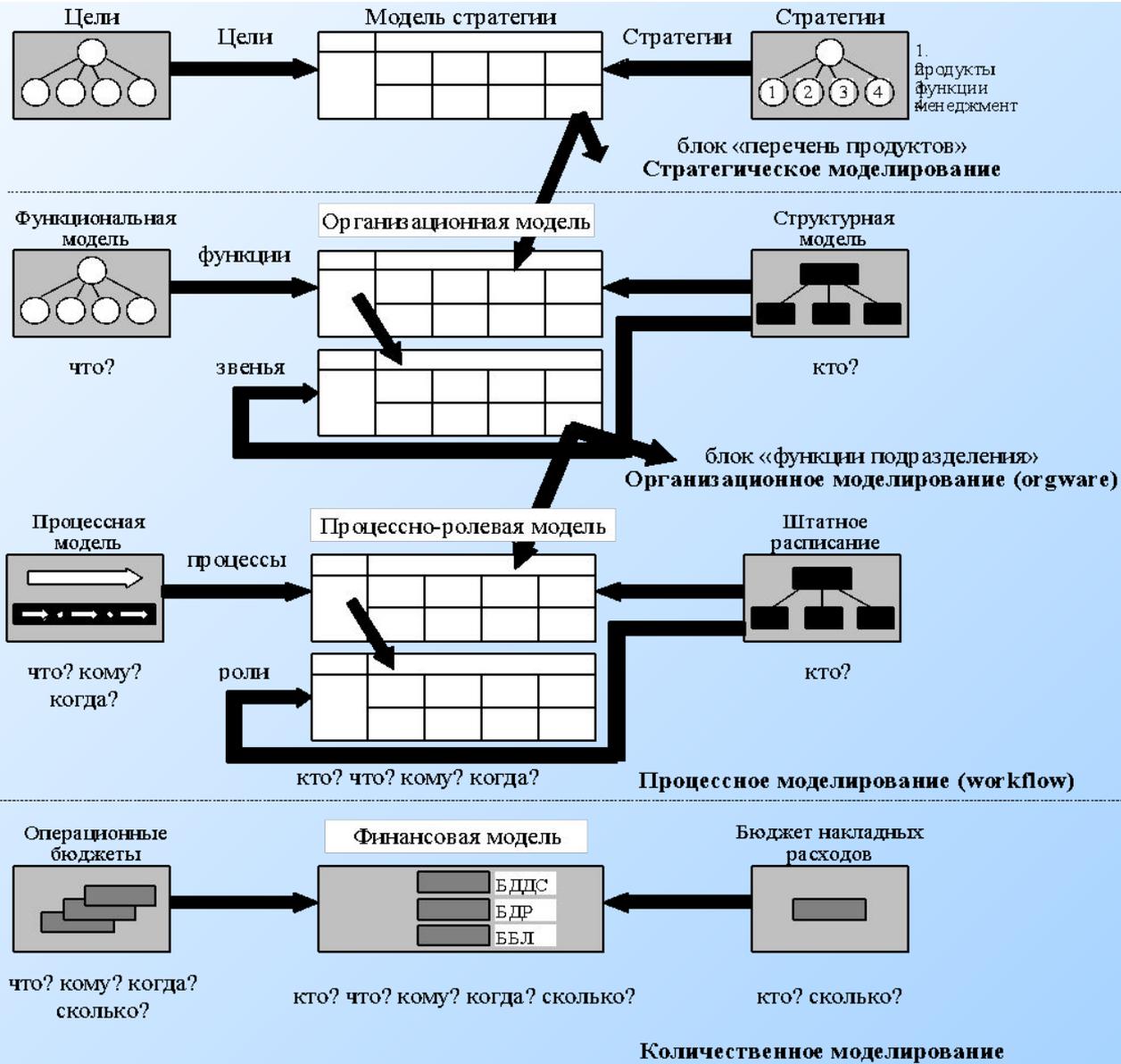
Инструментальные средства моделирования ЖЦ ПО

Под **комплексным (системным) моделированием (КМ) сложных объектов (СЛО)** любой природы (естественных, искусственных, реально существующих и виртуальных и т.п.) будем понимать методологию и технологии полимодельного описания указанных объектов, а также комбинированного использования методов, алгоритмов и методик многокритериального анализа, синтеза и выбора наиболее предпочтительных управленческих решений, связанных с созданием, использованием и развитием рассматриваемых объектов в различных условиях динамически изменяющейся внешней и внутренней обстановок

Инструментальные средства моделирования ЖЦ ПО

- проблема структурно-функционального синтеза облика полимодельного комплекса;
- проблема глубинного (интегративного) согласования используемых при комплексном моделировании Сло методов, моделей и алгоритмов;
- проблема параметрической и структурной адаптации полимодельного комплекса
- проблема верификации и валидации полимодельного комплекса;
- проблема автоматизации процесса комплексного моделирования Сло.

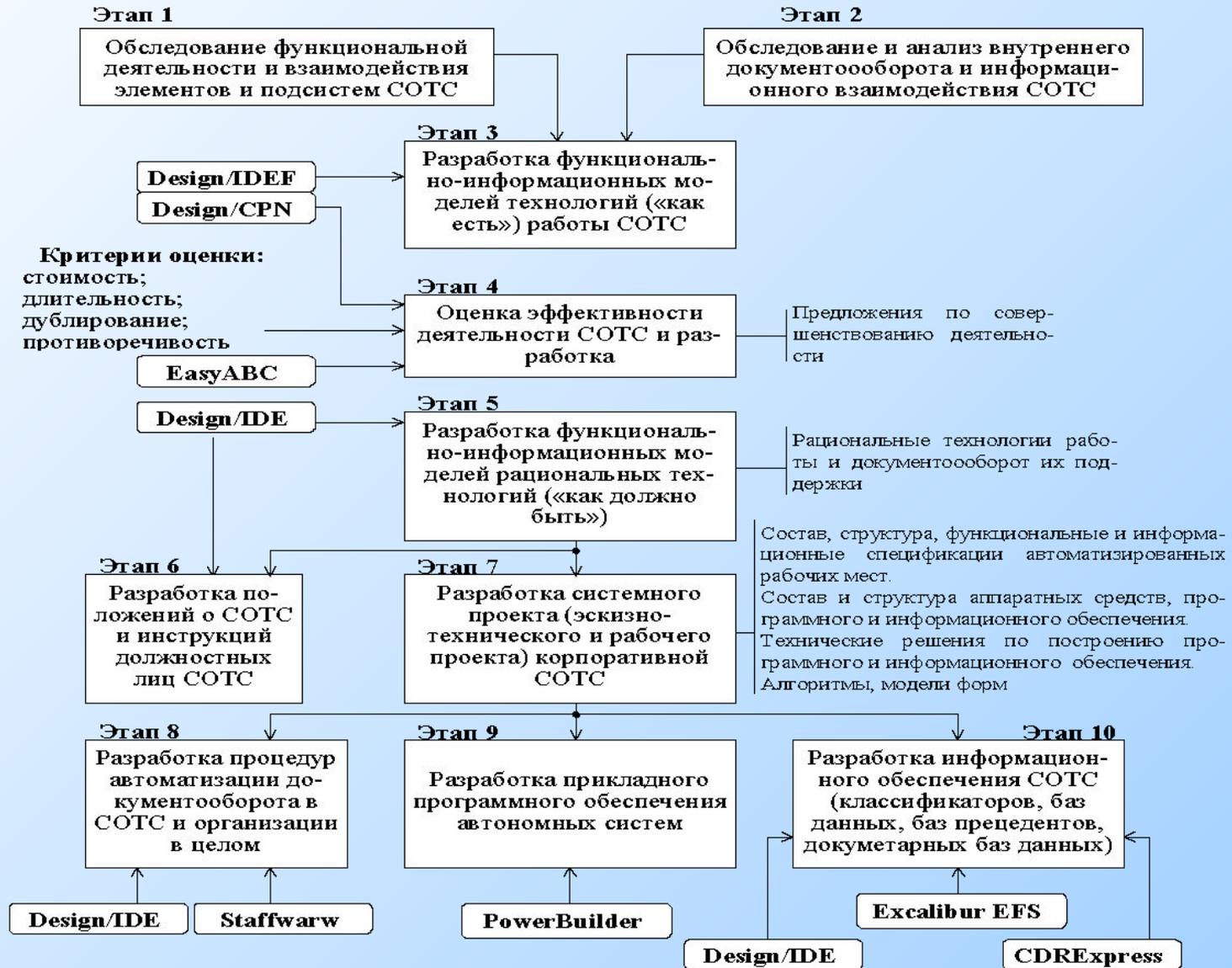
Инструментальные средства моделирования ЖЦ ПО



Инструментальные средства моделирования ЖЦ ПО



Инструментальные средства моделирования ЖЦ ПО



ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РАСЧЕТА И МНОГОКРИТЕРИАЛЬНОГО ОЦЕНИВАНИЯ ОСНОВНЫХ ХАРАКТЕРИСТИК И ПОКАЗАТЕЛЕЙ КАЧЕСТВА ФУНКЦИОНИРОВАНИЯ АСУ КА В ШТАТНЫХ И ЗАДАННЫХ УСЛОВИЯХ РАБОТЫ

Программный модуль 1 «Многокритериальность»	Программный модуль 2 «Надежность»	Программный модуль 3 «Расписание»	Программный модуль 4 «Устойчивость»	Программный модуль 5 «Пропускная способность»	Программный модуль 6 «Эффективность»
Программный модуль многокритериального анализа и упорядочения вариантов функционирования АСУ КА при различных сценариях изменения обстановки и воздействий	Программный модуль расчета и многокритериального анализа показателей структурной надежности и устойчивости АСУ КА	Программный модуль расчета расписания функционирования НКУ КА, а также расчета показателей пропускной способности и ресурсоемкости АСУ КА для детерминированных сценариев изменения внешних воздействий	Программный модуль расчета и оптимизации показателей робастности и динамической устойчивости программ функционирования АСУ КА для интервально заданных сценариев изменения внешних воздействий	Программный модуль расчета показателей пропускной способности и ресурсоемкости АСУ КА для стохастических сценариев изменения внешних воздействий	Программный модуль расчета показателей эффективности применения АСУ КА для стохастических сценариев изменения внешних воздействий

**Реализация комплекса аналитико-имитационных моделей подсистем АСУ КА
в разработанных программных модулях**

№ п/п	Наименование моделей подсистем АСУ КА	Реализованные модели в составе программных модулей									
		Модуль «Надежность»		Модуль «Расписание»		Модуль «Устойчивость »		Модуль «Пропускная способность»		Модуль «Эффектив- ность»	
		АМ	ИМ	АМ	ИМ	АМ	ИМ	АМ	ИМ	АМ	ИМ
1.	АИМ тракта ТМИ	+	-	+	-	+	-	-	+	+	-
2.	АИМ тракта ИТНП	+	+	+	-	+	-	-	+	+	-
3.	АИМ тракта КПИ	+	-	+	-	+	-	-	+	+	-
4.	АИМ тракта Спи	+	-	+	-	+	-	-	+	+	-
5.	АИМ ЦУП КА	+	-	+	-	+	+	-	+	+	-
6.	АИМ ИП ОД	+	-	+	-	+	-	-	+	+	-
7.	АИМ внешней среды	+	+	-	-	+	+	-	+	+	-

Концептуальная модель взаимодействия Бизнеса с ИТ службой на основе сервисно-ориентированного подхода

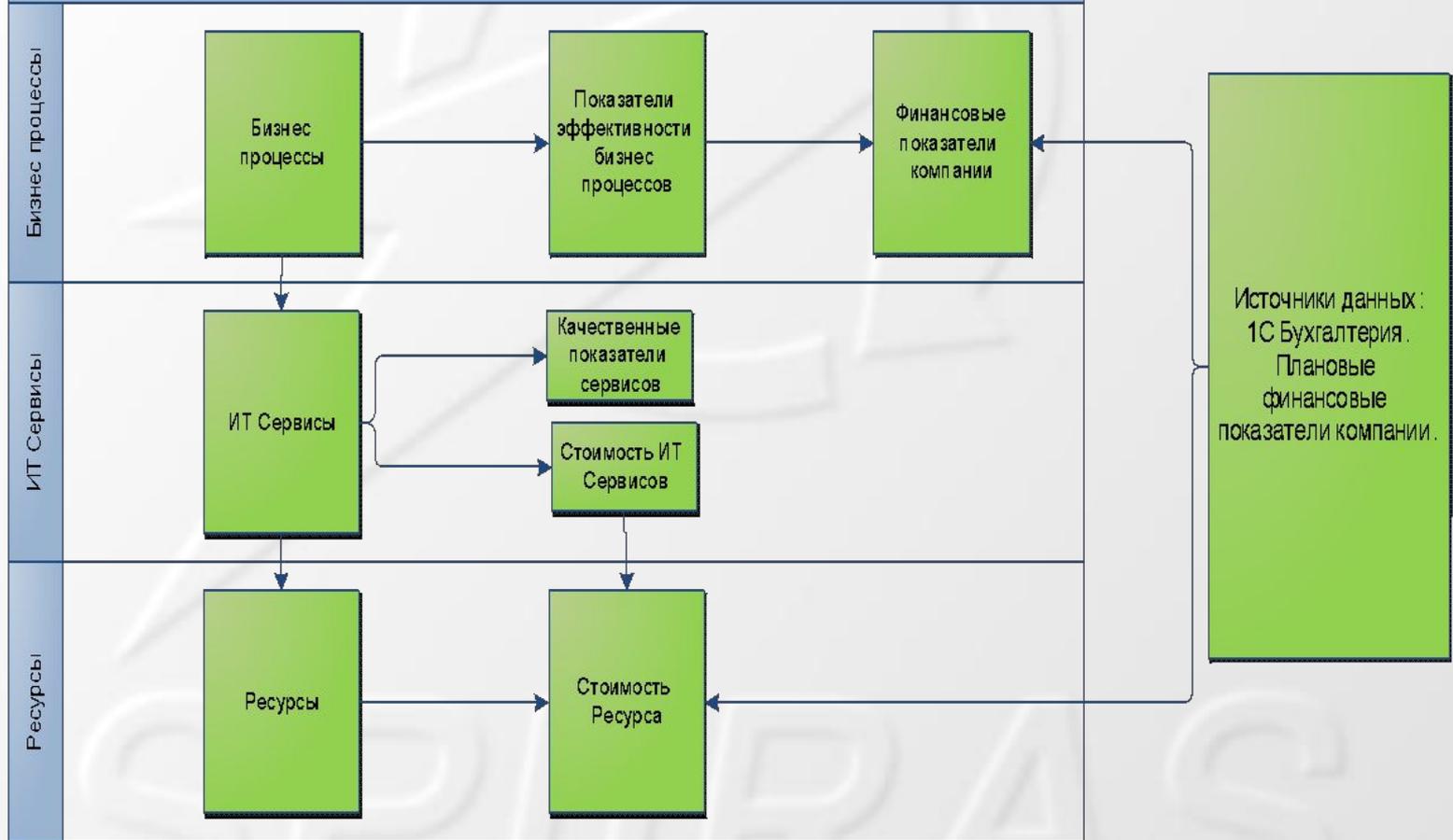
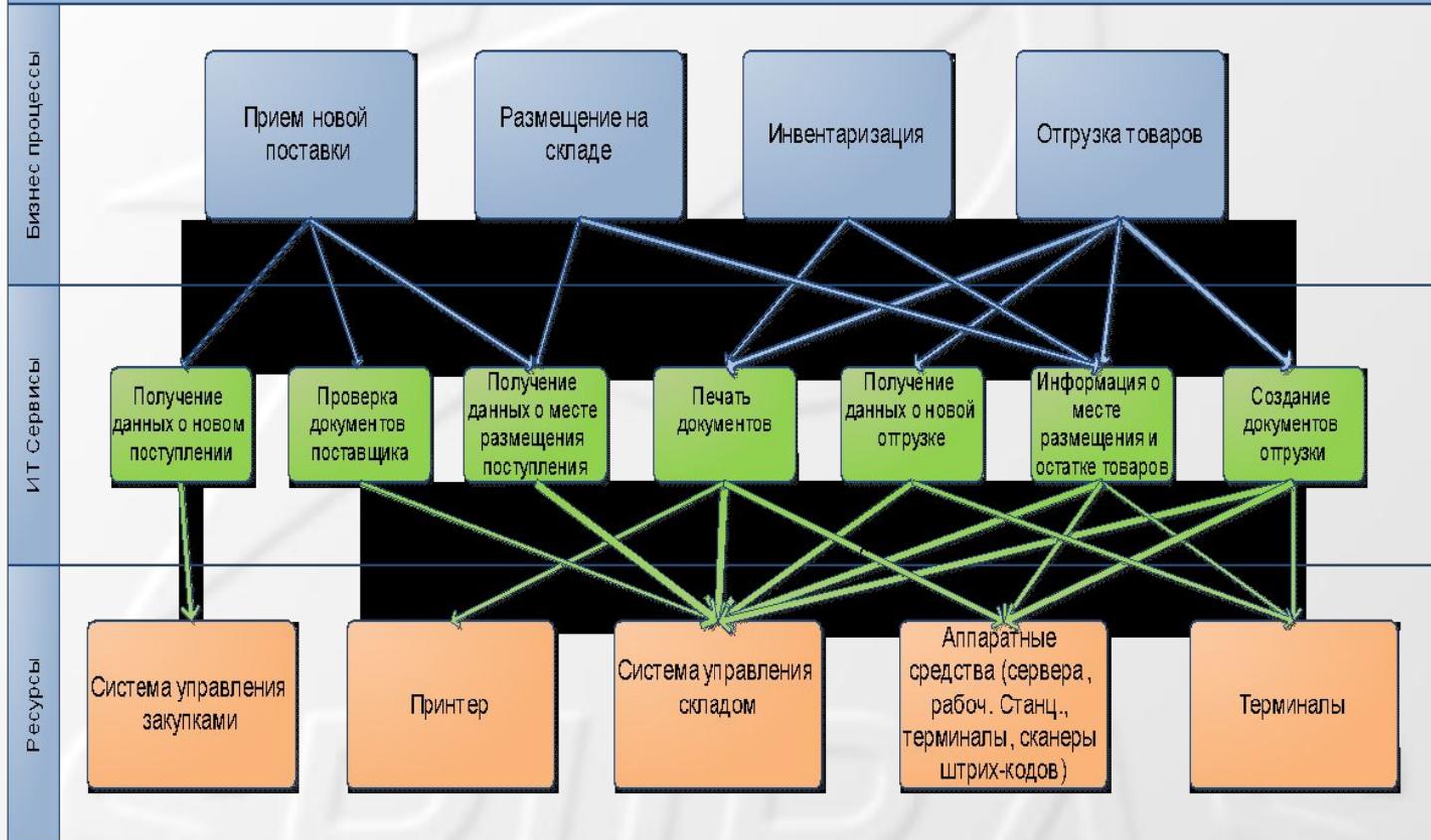


Схема взаимосвязи Бизнеса с ИТ сервисами: На примере системы управления складом

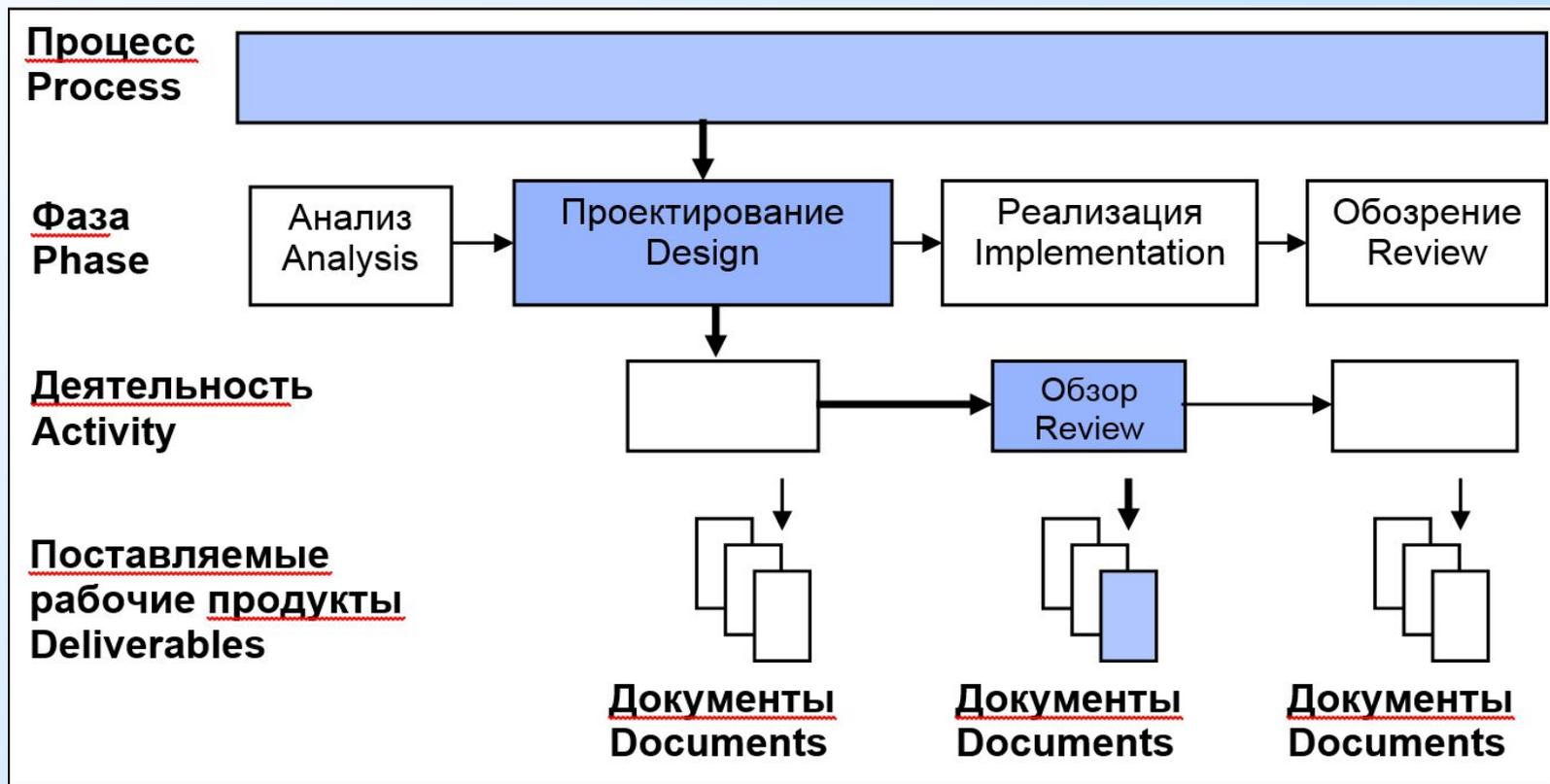


Задание 2



- Выберите какой-либо известный Вам проект
- Определите тип работы в нем: срочное исправление ошибок, тестирование, проектирование, и т.д.
- Используя матрицу выбора ЖЦ, примите решение, какой ЖЦ или комбинация ЖЦ наиболее подходит для Вашего проекта и объясните почему

Обобщенная схема повторяемого процесса разработки



Общая схема повторяемого процесса разработки ПО - Generic Repeatable Software Process

