ISO/IEC 12207
IEEE/EIA 12207
Software Life Cycle Processes
Supporting Life Cycle Processes

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ
CASE-технологии

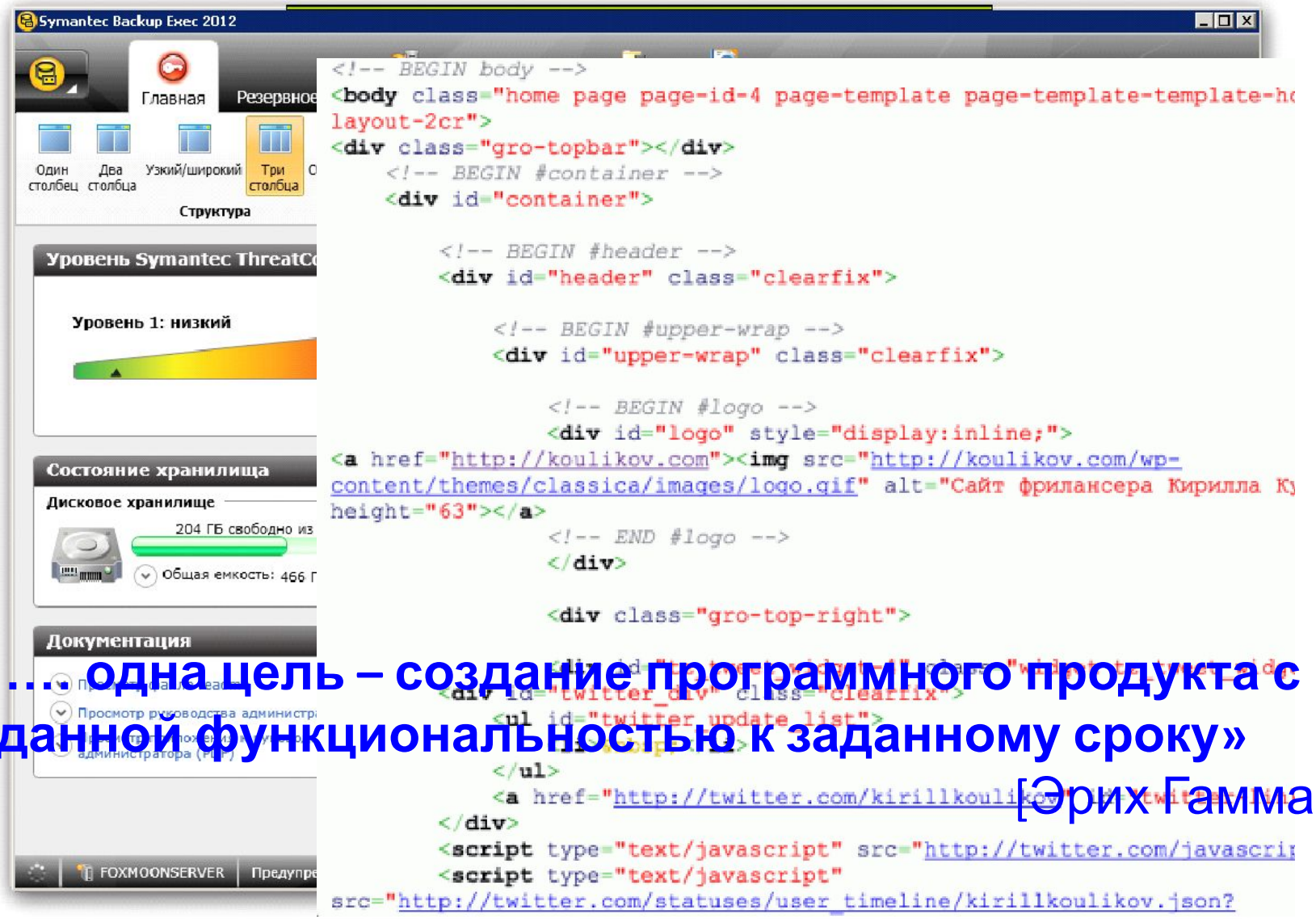
Лекции – 18 часов
Лабораторный практикум – 18часов
Домашнее задание – 10 часов
Зачет

канд. техн. наук, доцент
Андиева Елена Юрьевна
55_elena@mail.ru

ISO/IEC 12207
IEEE/EIA 12207
Software Life Cycle Processes
Supporting Life Cycle Processes

Л_1_Введение

- Предпосылки и история
  Технологии разработки и анализа ПС
  Методы и инструментальные средства ПИ
  Языки и системы программирования, семантика программ
  Методы, средства и системы управления базами данных и знаний
  Актуальность
  Основные понятия и определения
  Принципы программной инженерии



«…… одна цель – создание программного продукта с заданной функциональностью к заданному сроку»

[Эрих Гамма]

SOFTWARE ENGINEERING

Закон Мерфи: если существуют два способа сделать что-либо, причём один из которых ведёт к катастрофе, то кто-нибудь изберёт именно этот способ.

Следствия:

1. Всё не так легко, как кажется...
2. **Всякая работа требует больше времени, чем вы думаете.**
3. **Из всех возможных неприятностей произойдёт именно та, ущерб от которой больше.**
4. Если четыре причины возможных неприятностей заранее устранены, то всегда найдётся пятая.
5. **Предоставленные сами себе события имеют тенденцию развиваться от плохого к худшему.**
6. **Как только вы принимаетесь делать какую-то работу, находится другая, которую надо сделать ещё раньше.**
7. **Всякое решение плодит новые проблемы.**

Проблемы ИТ проектов:

1. **Технические проблемы;**
2. **Проблемы «Человеческого фактора»;**
3. **Проблема «плохого» IT-решения;**
4. **Проблема взаимодействия компании-заказчика и разра**

Предпосылки и история

1. Первая ЭВМ **ENIAC** была создана под руководством **Джона Моучли в 1946 году** в Пенсильванском университете (г. Филадельфия).
2. В СССР первая ЭВМ **МЭСМ** была разработана под руководством **С. А. Лебедева** в Киеве в **1951 году**.
3. **В 1953**, в Москве разработана под руководством **С.А. Лебедева** самая быстрая ЭВМ в Европе того времени – **БЭСМ**, в ИТМ и ВТ (Институт точной механики и вычислительной техники АН СССР).
4. В США **в 1951 году Д. Моучли** запустил в серию **ЭВМ UNIVAC** (было выпущено 47 штук).
5. В СССР первой серийной ЭВМ была **БЭСМ-2** (67 штук), которая выпускалась с **1958 года**, практически сразу же за БЭСМ-2 была запущена в серию М-20, также серийно выпускались **ЭВМ Урал-1** (183 штуки, начиная с **1957 года**), с **1960 года** – **Минск-1** (220 шт.).
6. **В 1957 году** появился **язык FORTRAN (FORmula TRANslator),** созданный под руководством **Джона Бэкуса** в IBM. Те люди, которые собирали первые библиотеки полезных программ, придумали FORTRAN и реализовали первые трансляторы с него в коды ЭВМ, **создали новую науку – системное программирование.**

Предпосылки и история

7. В СССР системное программирование начало развиваться очень рано. Профессор МГУ А.А. Ляпунов ещё в 1953 году начал работы по операторному методу в программировании, в 1953-54 гг. А.А. Ляпунов читал лекции по программированию будущему академику А.П. Ершову, среди его учеников были А.И.Китов и Н.А.Криницкий, И.В. Поттосин и многие другие известные программисты.
8. **В 1956 году** были опубликованы **работы Ю.И. Янова по схемам программ**, чуть позже **работы С.С. Лаврова и А.П. Ершова по экономии памяти на основе раскраски графов.** Работы Ю.И. Янова, С.С. Лаврова и А.П. Ершова во всем мире признаны классическими, заложившими **основы теоретического программирования.**
9. В 1958 всемирно известный статистик Джон Тьюкей (John Tukey) впервые ввел **термин software – программное обеспечение.**

Предпосылки и история

10. **В конце 60-х – начале 70-х годов** прошлого века произошло событие, которое вошло в историю как **первый кризис программирования.**
11. Термин **software engineering (программная инженерия)** - впервые был озвучен **в октябре 1968 года**.
12. **В 1970 г. У.У. Ройс** произвел идентификацию нескольких стадий в ЖЦ ПО и было высказано предположение, что контроль выполнения стадий приведет к повышению качества ПО и сокращению стоимости разработки.
13. **В 1972 году IEEE** (Computer Society of the Institute for Electrical and Electronic Engineers, IEEE Computer Society – IEEE-CS (Компьютерное Общество) или IEEE) выпустил первый номер «Transactions on Software Engineering» – **Труды по Программной Инженерии.**
14. Первый целостный взгляд на эту область профессиональной деятельности появился **1979 году**, когда **Компьютерное Общество IEEE** подготовило стандарт **IEEE Std 730** по качеству программного обеспечения.
15. После 7 лет работы, **в 1986 году IEEE выпустило IEEE Std 1002 «Taxonomy of Software Engineering Standards».**

Предпосылки и история

16. В России термин **«технология промышленного программирования»** привнес капитан 1-го ранга профессор **В.П. Морозов в конце 1980 года.**

«Поначалу такие вещи, как отчуждение программы от её автора, контроль за ходом разработки, графические схемы ситуаций мы называли «шпионскими штучками», но постепенно пришло понимание, что работа коллективов из сотен программистов без таких средств невозможна. Поскольку в западных странах «площадь соприкосновения с пользователем» (выражение А.П. Ершова) у программирования была несравненно больше, там понимание особенностей промышленного программирования пришло несколько раньше».

«Более 20 лет использование терминов Computer Science и Software Engineering было полностью делом индивидуального вкуса».

[Что такое программная инженерия. Зав. каф. системного программирования СПбГУ Генеральный директор ЗАО «Ланит-Терком» доктор физ-мат наук, профессор А.Н. Терехов]

Предпосылки и история

17. **В конце 90-х годов XX века** знания и опыт, которые были накоплены в индустрии программного обеспечения за предшествующие 30-35 лет, а также более чем 15-летние попытки применения различных моделей разработки, – все это, наконец, оформилось в то, что принято называть дисциплиной программной инженерии – **Software Engineering**.

18. **В 1990 году** началось **планирование всеобъемлющих международных стандартов**, в основу которых легли концепции и взгляды стандарта IEEE Std 1074 и результатов работы образованной в 1987 году совместной комиссии ISO/IEC JTC 1 (ISO – International Organization for Standardization; IEC – International Electrotechnical Commission; JTC 1 – Joint Technical Committee 1).

Предпосылки и история

19. **В 1995 году** группа этой комиссии SC7 «Software Engineering» выпустила первую версию международного стандарта **ISO/IEC 12207 «Software Lifecycle Processes»**. Этот стандарт стал первым опытом создания единого общего взгляда на программную инженерию. Соответствующий национальный стандарт России – ГОСТ Р ИСО/МЭК 12207-99 [ГОСТ 12207, 1999] содержит полный аутентичный перевод текста международного стандарта ISO/IEC 12207-95 (1995 года).

20. В свою очередь, IEEE и ACM (ACM – Association of Computer Machinery), начав совместные работы еще в 1993 году с кодекса этики и профессиональной практики в данной области (ACM/IEEE-CS Code of Ethics and Professional Practice), **к 2004 году** сформулировали **два ключевых описания Software Engineering.**

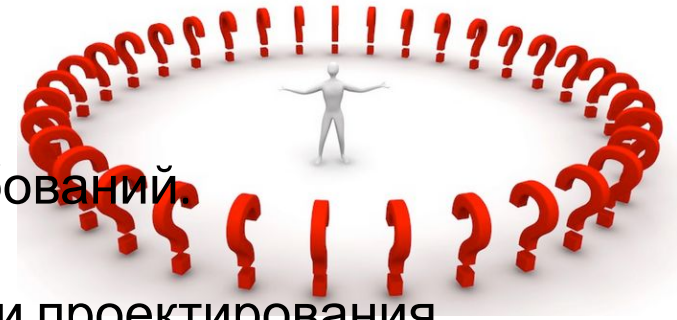
Предпосылки и история

21. **Guide to the Software Engineering Body of Knowledge (SWEBOK),** IEEE 2004 Version – Руководство к Своду Знаний по Программной Инженерии, в дальнейшем просто **«SWEBOK».**

22. **Software Engineering 2004 –** Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering – Учебный План для Преподавания Программной Инженерии в ВУЗах (данное название на русском языке представлено в вольном смысловом переводе).

http://www.computer-museum.ru

Технологии разработки и анализа ПС



1. Извлечение, анализ и моделирование требований.
2. Парадигмы моделирования ПС.
3. Проектирование ПО: концепции и стратегии проектирования, проектирование человеко-машинного интерфейса; готовых компонентов и методов их генерации; средства поддержки проектирования – инструментальных средств поддержки.
4. Статический и статико-динамический анализ.
5. Верификация и аттестация: основы, рецензия кода, тестирование, оценка пользовательского интерфейса, анализ проблем.
6. Динамическая верификация, интеграция различных методов верификации.
7. Метрики тестового покрытия.
8. Моделирование, измерение и тестирование производительности и потребления ресурсов.
9. Вопросы обучения технологиями разработки и анализа программ.

Методы и инструментальные средства ПИ

1. Методы и инструментальные средства извлечение, анализ и моделирование требований, проектирования, разработки и тестирования ПС.
2. Методы и инструментальные средства обеспечения информационной безопасности.
3. Методы и инструментальные средства управления надежностью ПС и рисками неблагоприятных событий на всех этапах их ЖЦ.
4. Методы и инструментальные средства верификации и валидации функциональных свойств и качества ПП.
5. Инструментальные средства документирования процессов разработки, сопровождения и модификации ПО.
6. Инструментальные средства поддержки системы управления качеством сложных ПП.
7. Методы и средства автоматизации процессов управления проектами разработки сложных ПС.

Языки и системы программирования, семантика программ

Парадигмы, языки и системы программирования.

1. Формальные модели и программные механизмы представления синтаксиса и семантики языков программирования.
2. Средства и системы компиляции программ.
3. Опыт использования современных языковых средств и систем программирования в различных предметных областях.

императивное	программа = последовательность действий, связанных условными и безусловными переходами
процедурное	программа = последовательность процедур, каждая из которых есть последовательность элементарных действий и вызовов процедур, структурированных с помощью структурных операторов `if`, `for` и `while`
объектно-ориентированное	программа = несколько взаимодействующих объектов, функциональность (действия) и данные распределяются между этими объектами
функциональное	прогамма = система определений функций, описание того, что нужно вычислить, а как это сделать — решает транслятор; последовательность действий не прослеживается
продукционное (логическое)	программа = система определений и правил вида "условие => новый факт"
сентенциальное	программа = система правил вида "шаблон => трансформирующее действие"
событийное	программа = система правил вида "событие => новые события" + диспетчер событий
автоматное	программа = конечный автомат или автомат специального типа

Методы, средства и системы управления базами данных и знаний

1. Модели, методы и средства формального описания знаний, хранения и управления ими.
2. Модели, методы и средства управления распределёнными данными, в том числе из источников с разной структурой их организации.
3. Системы и механизмы управления базами данных.
4. Опыт использования программного обеспечения баз данных и знаний в различных прикладных областях.
5. Подготовка кадров в области ПО современных систем управления данными и знаниями.

Актуальность программной инженерии

ISO/IEC 12207
IEEE/EIA 12207
Software Life Cycle Processes
Supporting Life Cycle Processes

- 80% проектов в результате - не вполне удовлетворяют бизнес-пользователя
- 75% менеджеров ИТ проектов обвиняют бизнес-заказчика в срыве сроков и бюджета из-за постоянных уточнений требований

http://www.pmi.ru/profes/16022010_Ovchinnikova.pdf

Актуальность программной инженерии

ISO/IEC 12207
IEEE/EIA 12207
Software Life Cycle Processes
Supporting Life Cycle Processes

- 95% компаний жалуются на саботаж пользователей изменениям в работе
- 80% имеют проблемы со сдачей в промышленную эксплуатацию
- 60% жалуются на проблемы с согласованием требований по проекту между бизнес-заказчиками
- 50% жалуются на недостаточное внимание руководства к ИТ проектам
- 70% систем сдаются с длинным перечнем недоработок
- 70% проектов не имеют плана опытной эксплуатации

http://www.pmi.ru/profes/16022010_Ovchinnikova.pdf
https://www.ibm.com/developerworks/ru/library/r-requirements/

SOFTWARE ENGINEERING

Актуальность программной инженерии

ISO/IEC 12207
IEEE/EIA 12207
Software Life Cycle Processes
Supporting Life Cycle Processes

Проблемы аутсорсинга: Service Level Agreement, SLA
Соглашение об уровне предоставления услуги

- 90% ТЗ не содержат проекты SLA
- 90% эксплуатационной документации не содержит норм и подробных инструкций по обслуживанию
- 85% процентов проектных команд не имеют квалификации в области сервисного обслуживания
- Бюджет 80% проектов не содержит расходов на организацию сервиса

http://www.pmi.ru/profes/16022010_Ovchinnikova.pdf

SOFTWARE ENGINEERING

Основные понятия и определения

Программная инженерия (Software Engineering):

Систематическое применение научных и технических знаний, методов и опыта для разработки, реализации, тестирования и документирования программного обеспечения [ISO/IEC 2382-1].

Применение систематизированного, упорядоченного, количественно измеримого подхода к разработке, эксплуатации и сопровождению программного обеспечения, что означает **применение инженерии к программному обеспечению** [ISO/IEC 24765].

1. Теория, основанная на опыте.
2. Теория, созданная отдельно от практики.
3. От прикладных методов к теории.

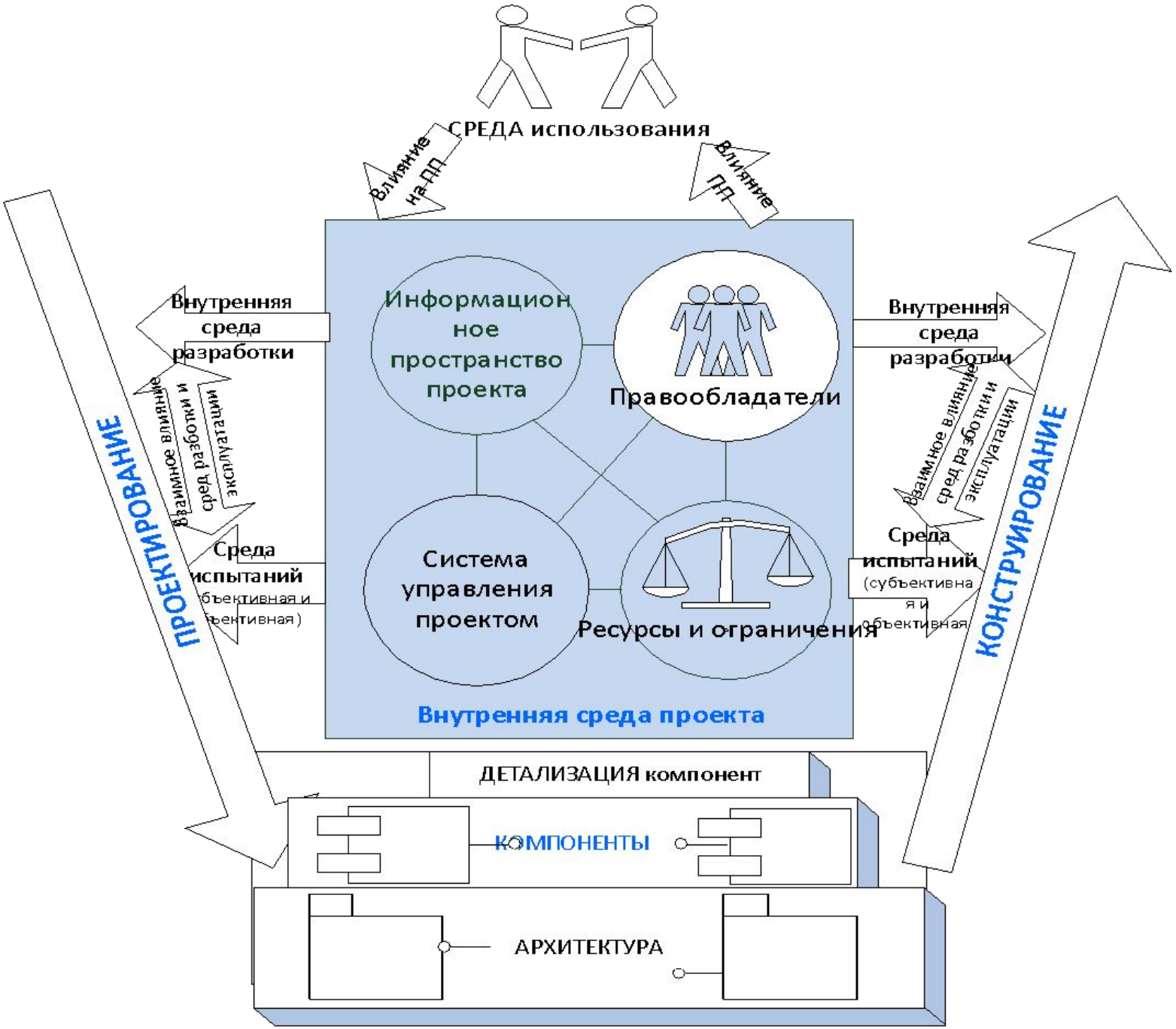
Основные понятия и определения

Важнейшие области:

1. Бизнес-процессы и оценка функционирования (Business Processes and Operational Assessment (BPOA))
2. Моделирование и Анализ (Modeling, Simulation, & Analysis (MS&A))
3. Архитектура систем/решений/тестирования (System/Solution/Test Architecture (SSTA))
4. Анализ стоимости жизненного цикла и соотношения прибылей и затрат (Life Cycle Cost & Cost-Benefit Analysis (LCC & CBA))
5. Обеспечение пригодности к обслуживанию/ логистика (Serviceability/ Logistics (S/L))
6. Управление рисками/конфигурацией/исходным состоянием (Management: Risk, Configuration, Baseline)

http://www.ibm.com/developerworks/ru/library/r-temnenco/

SOFTWARE ENGINEERING


СРЕДА использования
Влияние на ДП
Влияние ПП
Внутренняя среда разработки
Взаимное влияние сред разработки и эксплуатации
Среда испытаний
(субъективная и объективная)
Информационное пространство проекта
Правообладатели
Система управления проектом
Ресурсы и ограничения
Внутренняя среда проекта
Внутренняя среда разработки
Взаимное влияние сред разработки и эксплуатации
Среда испытаний
(субъективная и объективная)
ПРОЕКТИРОВАНИЕ
КОНСТРУИРОВАНИЕ
ДЕТАЛИЗАЦИЯ компонент
КОМПОНЕНТЫ
АРХИТЕКТУРА

Принципы программной инженерии

1. Переход **от редукционистского к системному подходу.**
2. Переход **от структуры к поведению** («…что происходит в системе — это её изменения во времени, ...нужно рассматривать изменения во времени и взаимодействия, разворачиваемые во времени»).
3. Переход **от одной группы описаний ко множественности групп описаний.**
4. Переход **от рабочего проектирования** (модель – код) **к** обязательному предварительному **архитектурному.**

«Вместо того, чтобы сразу работать с «кодом» программ…и прочими реализационными описаниями систем, для начала нужно сделать сущностное, не зависящее от деталей реализации описание создаваемой системы: архитектуру. **Архитектура** описывает основные подсистемы и их взаимодействие в языке, свободном от деталей реализации.

Одной архитектуре может соответствовать **множество разных реализаций. Архитектура более живуча, чем её реализации»**.

Принципы программной инженерии

5. Переход **от непосредственной реализации к моделецентричной реализации** (все ошибки убираются на этапе моделирования, а не на этапе реального воплощения ...).
6. Переход **от документоцентризма к датацентризму** (… работа с изменениями должна вестись в терминах отдельных данных, а не «документов»).
7. Переход **от работы «для одного хозяина» к работе со множеством заинтересованных сторон**.
8. Переход **от «проверки» к раздельным верификации к валидации.**

Верификация – проверка на соответствие формальным требованиям.

Валидация — проверка того, что требования конкретного внешнего потребителя или пользователя продукта, услуги или системы могут быть удовлетворены.

Принципы программной инженерии

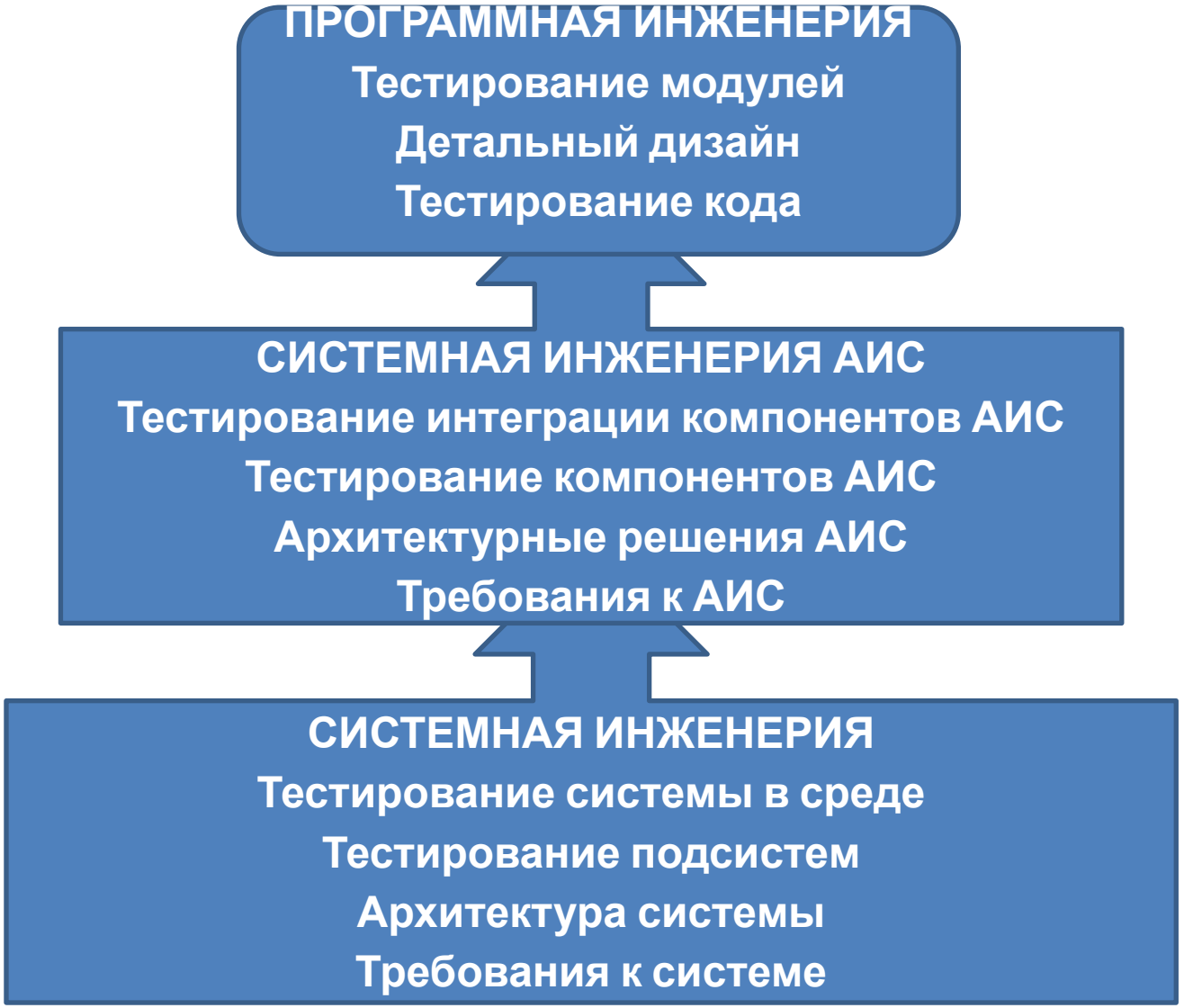
9. Переход **от методов жесткого планирования к использованию гибких методов.**
10. Переход **от «технологического конвейера» к «заказам-поставкам»** Процесс управления ЖЦ не «конвейер» - выполнение серии технологически предписанных работ (в теории управления организационными процессами это называют «оркестровка»… В жизни независимые действующие лица участвуют в процессе, выполняют контракт на оказание услуги по своей части – в теории управления организационными процессами это называют **«хореография».**

Платформа **Jazz.** Функциональность сервиса поддерживается многими продуктами **IBM Rational**, включая **Team Concert** для планирования проекта, мониторинга прогресса и отчетности: «От развития до развертывания ... и все между ними».

Powered by JazzHub

JazzHub - это процесс разработки ПО как сервис.
Университеты (преподаватели, студенты) могут использовать его бесплатно в образовательных целях.
Облачный сервис JazzHub: https://hub.jazz.net/

Принципы системной инженерии



САМОСТОЯТЕЛЬНО!

1) Системная инженерия. Историческая справка
2) Системная инженерия и традиционные инженерные дисциплины
3) Системная инженерия и управление проектом
4) Системы, нуждающиеся в системной инженерии

СПАСИБО ЗА ВНИМАНИЕ !!!

Вопросы….