


Реинжиниринг Программной инженерии

Петренко А.К., ИСП РАН,

www.ispras.ru/~petrenko



План

- «Программная инженерия»
 - Возникновение термина, наука/техническая дисциплина/отрасль промышленности, описание области исследований, «сопредельные территории»
 - Успехи и нерешенные проблемы
 - Конструктивные ответы на вызовы
 - Выводы
- 



Определения

- **(Wikipedia:)Software engineering (SE)** is a profession dedicated to designing, implementing, and modifying software so that it is of higher quality, more affordable, maintainable, and faster to build. It is a "systematic approach to the analysis, design, assessment, implementation, test, maintenance and reengineering of software, that is, the application of engineering to software."





Определения

- **(Wikipedia:)**Software engineering (SE) is a *profession* dedicated to designing, implementing, and modifying software so that it is of *higher quality, more affordable, maintainable, and faster to build*. It is a "systematic approach to the analysis, design, assessment, implementation, test, maintenance and reengineering of software, that is, the application of engineering to software."

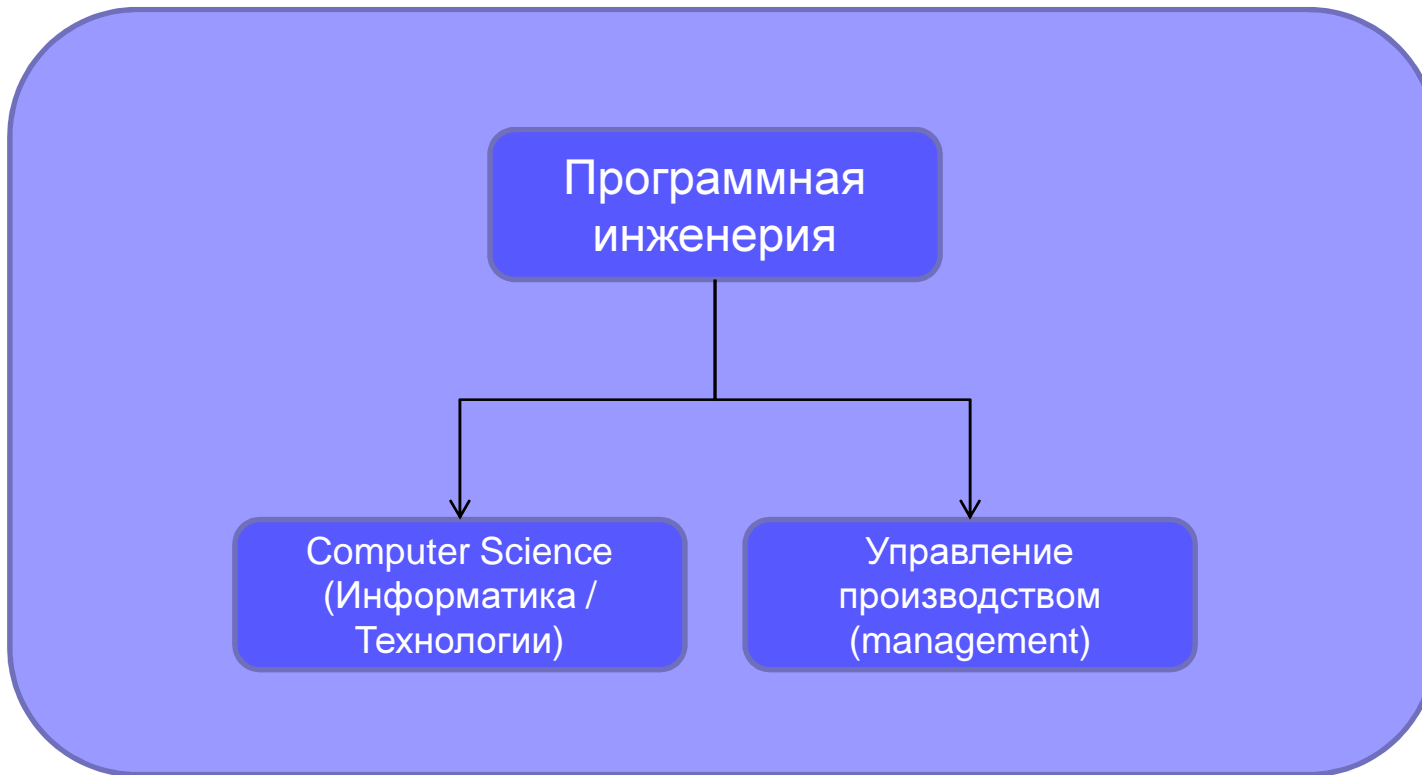


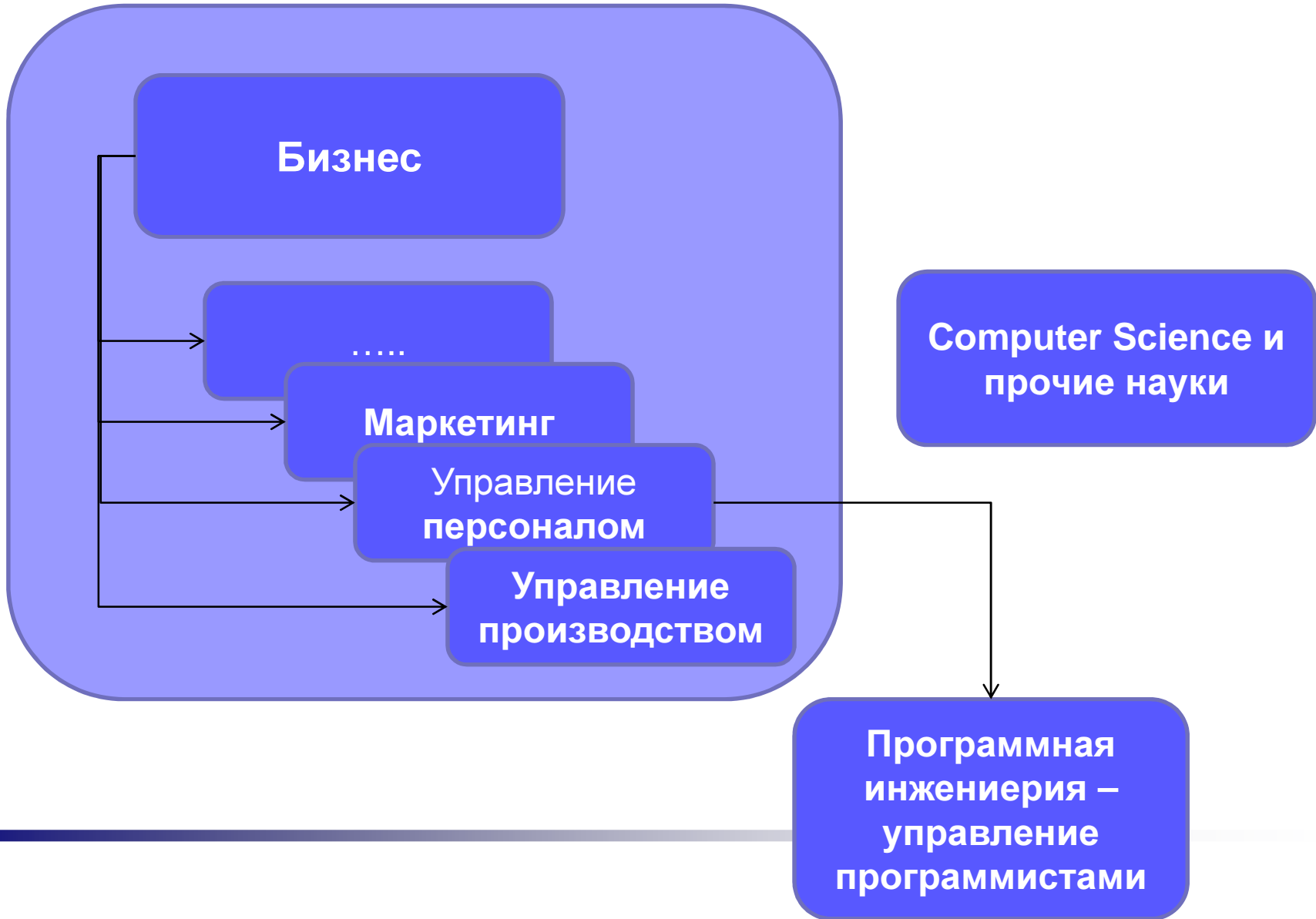
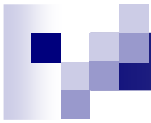


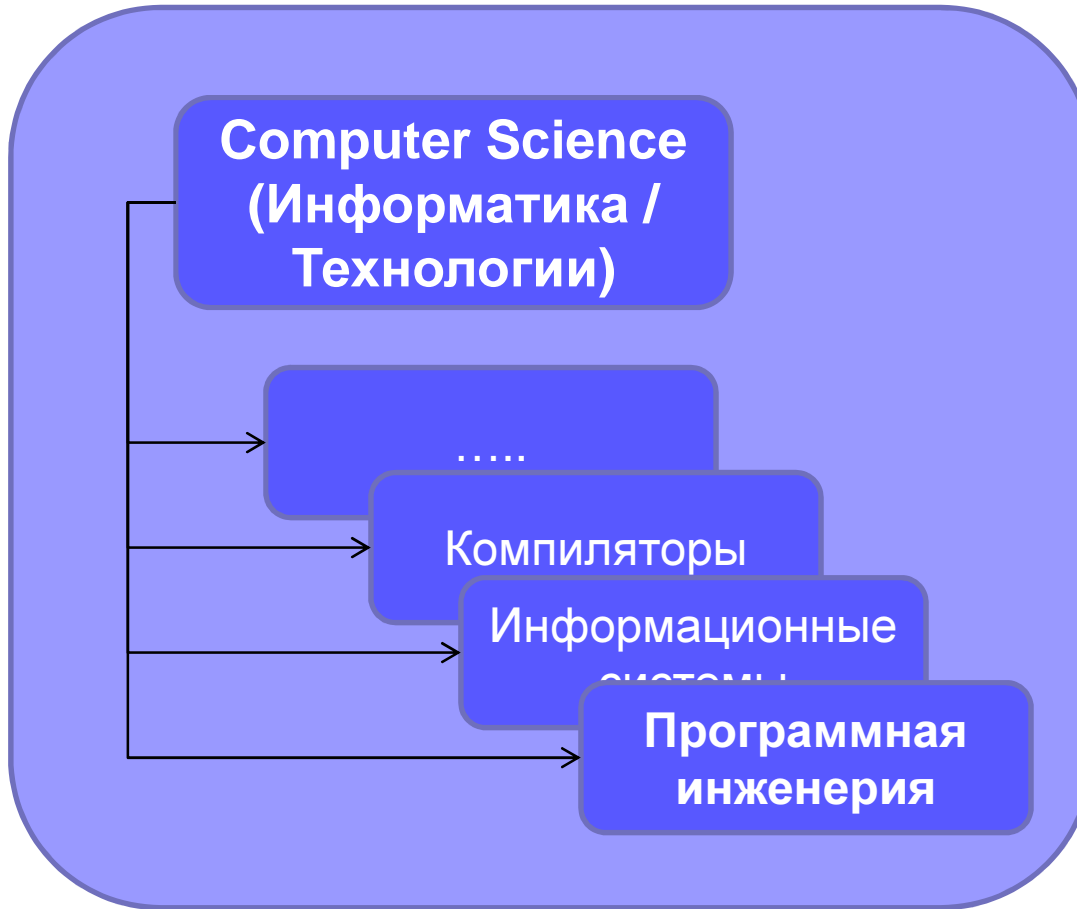
Определения

- **(Википедия:) Программная инженерия** — это интегрирование принципов [математики](#), [информатики](#) и [компьютерных наук](#) с [инженерными подходами](#), разработанными для производства осязаемых материальных артефактов. ([Curricula Recommendations Software Engineering SE 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering](#)).



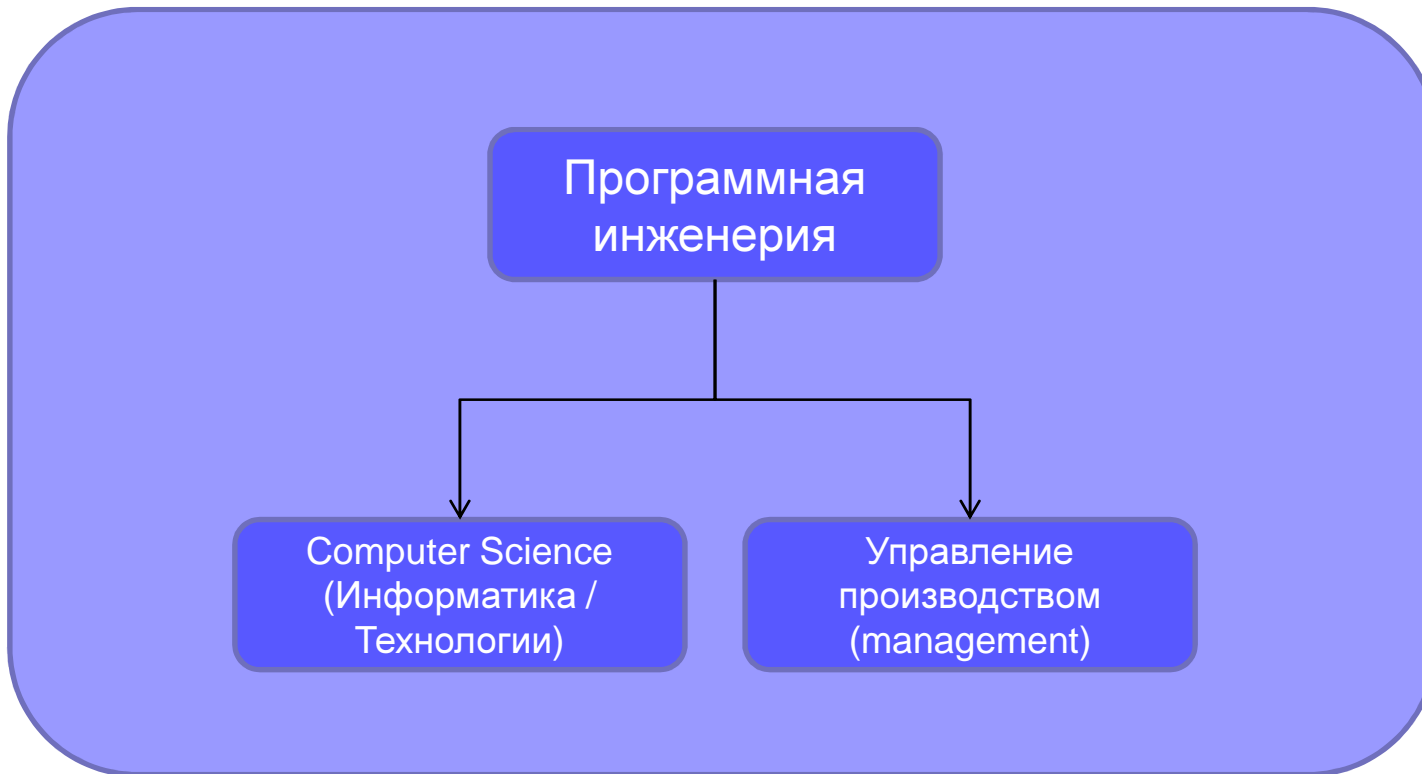








Управление
производством
(management)

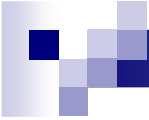








Миры программной инженерии

- **Научная дисциплина**
 - Общие закономерности, методы изучения, фундаментальные знания
 - **Инженерная дисциплина**
 - Технологии, методы адаптации, внедрения общих подходов в практику, вопросы организации и управления
 - **Отрасль промышленности**
 - Организация бизнеса (анализ рынка)
 - **Область профессиональной деятельности**
 - Профессиональные союзы и общества, сертификация инженеров
- 



Motivation. Парадоксы в развитии компьютерных/программных систем


- Безудержный рост размеров и сложности, при этом:
 - Достаточно высокий уровень надежности
 - Беспомощность перед проблемами создания программных систем без дефектов, в частности, гарантирующих информационную безопасность при полной доступности **всей имеющейся** информации о программах, данных и устройствах обработки и передачи информации (в отличие, например, от физики, химии, биологии – где многое еще не известно).
 - Нет научной методики оценки и прогнозов
 - Множество технологий разрабатываются и лишь единицы оказываются востребованными
- 



Уточнение рамок рассмотрения


- Технические аспекты программной инженерии – методы и инструменты разработки и анализа программных систем
- Общецелевые и специализированные (domain specific) технологии программирования.

Вне зоны рассмотрения:

- компиляторы, интегрированные среды разработки, СУБД, операционные системы, ПО промежуточного слоя, машинная графика, разработка игр, системы распознавания изображений, вычислительная математика и проч.
- 





Направления развития

- Сквозная инструментальная поддержка жизненного цикла создания и эксплуатации программных систем (ALM)
 - Собственно технологии \equiv Средства разработки и исследования программных систем.
 - Model driven/Model based подходы. Полярные трактовки моделирования:
 - Модель отделена от реализации (от программы)
 - Программа во всех фазах ее развития сама по себе является моделью.
 - «Идеологический» (мотивационный) фундамент
 - С чего начинать? «курица или яйцо».
 - В программной инженерии роль «курицы и яйца» выполняет треугольник «требование-реализация-тест»
 - Интероперабельность, совместимость программных систем, стандартизация интерфейсов
 - Корректность программ. Статический и динамический анализ программ
 - Проверка корректности использования машинных и языковых конструкций, стандартных интерфейсов аппаратуры и ОС
 - Проверка функциональных спецификаций
- 



Актуальные инициативы

- SEMAT (OMG) - Software Engineering Method and Theory
 - OSEE (Boeing) – Open System Engineering Environmet
 - ALM (IBM, HP, Borland) - Application lifecycle management
 - Microsoft:
 - Verifying Compiler, RiSE, interoperability@microsoft, Interoperability Bridges and Labs Center (Microsoft), Microsoft ALM solutions
 - MDA, MDE, CESAR, ModelBus, MODELPLEX
 - OSLC (IBM) - Open Services for Lifecycle Collaboration
 - Еще раз о SEMAT
- 



SEMAT - Software Engineering Method and Theory


- *Semat seeks to develop a rigorous, theoretically sound basis for software engineering practice, and its wide adoption by industry and academia.*
- *Troika* - Bertrand Meyer, Ivar Jacobson & Richard Soley





SEMAT манифест

Software engineering is gravely hampered today by immature practices. Specific problems include:

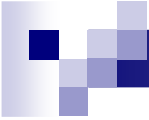
- The prevalence of fads more typical of fashion industry than of an engineering discipline.
 - The lack of a sound, widely accepted theoretical basis.
 - The huge number of methods and method variants, with differences little understood and artificially magnified.
 - The lack of credible experimental evaluation and validation.
 - The split between industry practice and academic research.
 - We support a process to refound software engineering based on a solid theory, proven principles and best practices that:
 - Include a kernel of widely-agreed elements, extensible for specific uses
 - Addresses both technology and people issues
 - Are supported by industry, academia, researchers and users
 - Support extension in the face of changing requirements and technology
- 



Semat tracks

- Architecture Track (led by Paul McMahon)
- Requirements Track (led by Dave Cuningham)
- Theory Track (led by Michael Goedicke)
- Kernel Language Track (led by Michael Goedicke)
- Universal Track (led by Ian Spence)
- Assessment Track (led by Paul McMahon)
- Definition Track (led by Mira Kajko-Mattsson)

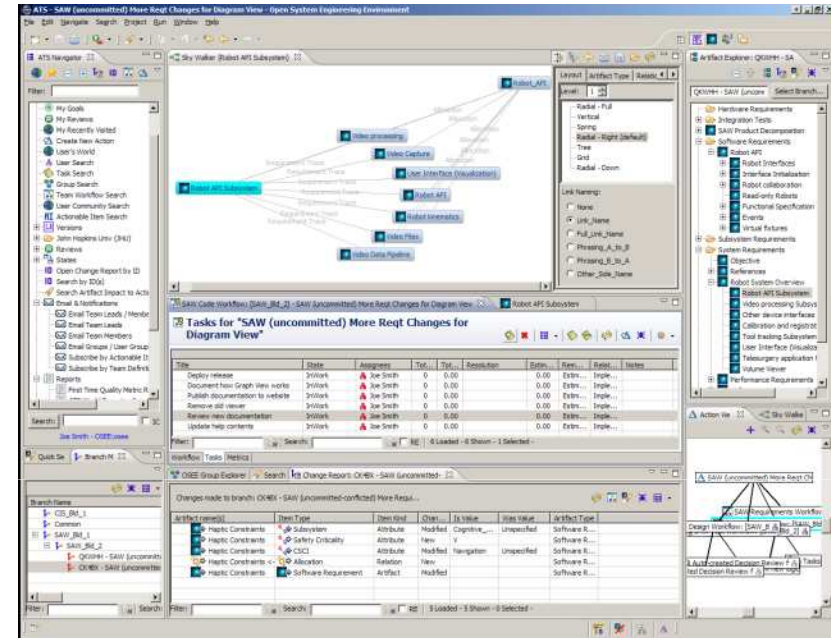
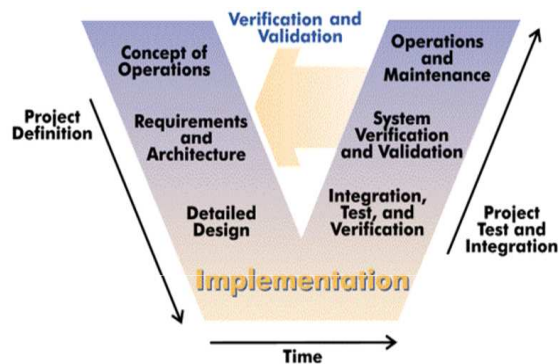




OSEE - Среда разработки Open System Engineering Environment

- Инициатор: Boeing
- База (framework): Eclipse
- Состав
 - Framework
 - ATS (Action Tracking System)
 - Coverage (Оценка полноты тестового покрытия)
 - OTE (Open Test Environment)

The Open System Engineering Environment is an integrated, extensible tool environment for large engineering projects



Extensible Data Model

All project data is stored in OSEE's user-definable data model, which offers revision control, bidirectional traceability, project status reporting, and metrics.

Full Lifecycle Support

A Potent Framework

In addition to excelling as a stand-alone application, OSEE offers a full-featured API, enabling developers to harness OSEE's powerful data management capabilities for their own applications.

Version and Variant Management

ALM - Application lifecycle management - IBM

Application lifecycle management

Manage your software development and delivery from definition to deployment



- Requirements management
- Project, process, and portfolio management
- Architecture management
- Software change, configuration, and release management
- Quality management

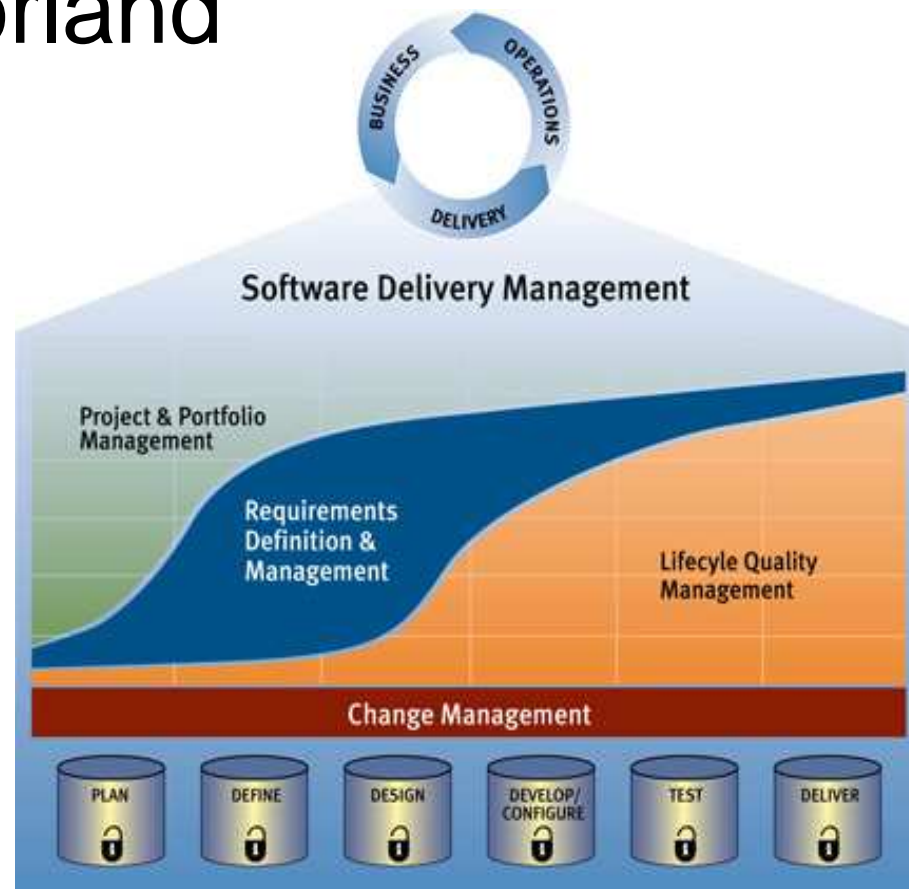
ALM - Application lifecycle management - HP



- HP Application Lifecycle Management
- HP Business Process Testing software
- HP Change Impact Testing for SAP Applications software
- HP Functional Testing software
- HP Quality Center software
- HP Requirements Management module
- HP Service Test Management module
- HP Service Test software
- HP Sprinter software
- HP Test Data Management Software
- HP Unified Functional Testing software
- Performance Validation
- Quality Assurance

ALM - Application lifecycle management - Borland

- Requirements Definition & Management
- Testing and ASQ
- Change Management





ALM - Application lifecycle management - Microsoft



- Architecture Explorer
- TFS -- Team Foundation Server
- Microsoft Test Manager
- Microsoft Test Runner



ALM - Application lifecycle management – Microsoft Research

Research in Software Engineering team (RiSE).


- **Professional Developer Tools**
- [AjaxView](#) Profiler for AJAX web sites.
- [CHESS](#) Systematic Concurrency Testing for .NET and Win32.
- [Code Contracts](#) Design by Contract for .NET.
- [Doloto](#) Time optimizer for web 2.0 applications.
- [Moles](#) Isolation Framework for unit testing for .NET.
- [Pex](#) Automated Whitebox Testing for .NET.
- **Open Source**
- [Boogie](#) Intermediate procedural Verification Language that produces verification conditions.
- [CCI](#) API that allows applications to efficiently analyze or modify .NET assemblies, modules, and debugging (PDB) files.
- [Nmodel](#) Source of Model-based Software Testing and Analysis Tool with C#.
- **Academic Tools and Open Source**
- [Spec#](#) Mechanical verifier and runtime checker for C#.
- [VCC](#) Mechanical verifier for concurrent C programs.
- **Academic Tools and Closed Source**
- [Z3](#) High-performance automatic (SMT) theorem prover.



MDA, MDE, CESAR, ModelBus, MODELPLEX

- MDA – Model Driven Architecture
- MDE – Model Driven Engineering
- CESAR, ModelBus, MODELPLEX проекты FP7
- **OSLC - Open Services for Lifecycle Collaboration (IBM)**






SEMAT – еще раз

- Начинать с анализа реальных практик





Выводы. Современные тенденции

- Унифицировать и механизировать процессы
 - Отказ от однонаправленных моделей жизненного цикла («водопадная» модель)
 - Средства интеграции разнородных инструментов поддержки ЖЦ
 - «Три источника» разработки ПО
 - Требования
 - Дизайн
 - Тесты
 - «Три составные части» ?
- 

Мы в начале пути

Лабиринт: помоги Ивану найти путь к царевне-лягушке

