

К теории программной инженерии

Состояние и направления развития

Б.А. Позин

д.т.н., профессор,

Председатель SEMAT Russian Chapter





Состояние индустрии

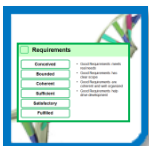


Специфические проблемы



Направления деятельности

- определение убедительной теоретической базы
- поиск ядра, состоящего из общепризнанных элементов



Использование ядра



Заключение

В двух словах – что нужно поменять



Производство

Профессионалы

Хотят стать экспертами. Задачи :
- **навыки и опыт нелегко снова использовать для нового продукта.**
- карьерный путь проходит по извилистой дороге от одной рекламной шумихи к другой.

Руководители

В больших компаниях существует много процессов.
Задачи :
- **Повторное использование практики**
- Повторное обучение
- “Повторное использование” людей
- **Эволюционные улучшения трудны**

Научная сфера

Образование

Обучение конкретным методологиям, таким как Scrum, RUP, вместо освоения универсальной методологической базы;
универсалы не появляются

Исследования

Разрыв между исследованием и производством
Нет общепризнанной теории

SEMAT ориентирует все программистское сообщество

Нас рассматривают как индустрию, заслуживающую доверия?



Blackberry users complain of fresh crash

COMMENTS (552)



BBC Radio 1 Newsbeat listeners described how the crash affected them

Sydney cash machine glitch gives customers extra money

Queues formed at cash machines in the Australian city of Sydney after they malfunctioned and began giving out large sums of money.

More than 40 Commonwealth Bank cash machines across the city were affected by the problem.

Officials said they were operating in stand-by



used by some of its

Незрелые практики сегодня серьезно препятствуют Программной Инженерии

INDEPENDENT

NEWS OPINION SP

UK World Business

News > UK > Home News

RBS admits
Monday as
financial ch

Week
ARE

base after software failure

Published: Aug 02, 2012 - 7:53 PM GMT

One of the largest US market makers is on the edge of bankruptcy after its poor performance on the stock market due to the technology failure.



by Dagmar Benesova
WBP Online Correspondent in Washington D.C.

Knight Capital Group, one of the biggest US market makers, said it lost \$440 million of the firm's capital in a trading disruption which messed up the market.

ment raised questions and concerns of

10:37 AM Forget Fack
10:19 AM BoE hopeful of econ
10:14 AM Oil falls amid US stock
09:07 AM Europe current account a
08:57 AM European stocks still up on
08:08 AM Euro climbs to 6-week high
07:58 AM Heineken in talks about
07:33 AM Europe stocks trade

Indices	
US500	1,412.15 -0.07%
US100	2,763.20 -0.09%
US30	13,216.7 -0.1%
EU50	2,440.0 -0.1%
UK100	5,700.0 -0.1%
DE30	7,000.0 -0.1%





Состояние индустрии

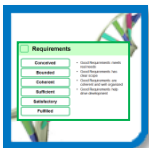


Специфические проблемы



Направления деятельности

- определение убедительной теоретической базы
- поиск ядра, состоящего из общепризнанных элементов



Использование ядра



Заключение

Программная индустрия подвержена МОДНЫМ ВЕЯНИЯМ



Двадцать лет назад программная инженерия развивалась “под эгидой” объектно-ориентированного программирования

15 лет назад популярными стали RUP и UML

12 лет назад получила массовое признание во всем мире модель CMMI

Несколько лет назад - экстремальное программирование XP

Сегодня пользуются спросом agile-практики наподобие Scrum, Lean and Kanban

Что будет завтра?



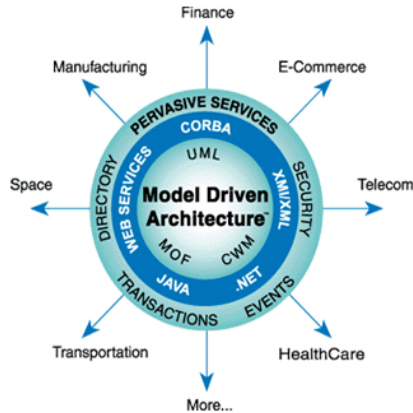
Это все хорошо, но это еще не все, что вам нужно!

Каждое такое модное течение хорошо в отдельности, но продолжает отсутствовать общая база, позволяющая сказать, что хорошо, а что плохо в проекте.

Каждый из нас знает,
как разработать **свое**
собственное
программное
обеспечение, но как у
сообщества, у нас
нет общей основы



Огромное количество методов...



- Все они по-своему уникальны и несравнимы

- Мы не можем выбрать лучшие практики для очередного проекта

- Мы не знаем, какие методы используются в большой компании

- Мы так и не знаем как успешно переносить накопленные знания от проекта к проекту.



Отсутствие заслуживающих доверия экспериментальных оценок и проверок



**Мы отказываемся от нашего старого метода и
заменяем его новым популярным,
не понимая полностью последствий**



**Практика
индустрии**



**Научные
исследования**

**Сохраняется разрыв между академическими исследованиями и
прикладными разработками**



Состояние индустрии

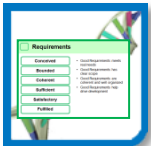


Специфические проблемы



Направления деятельности

- определение убедительной теоретической базы
- поиск ядра, состоящего из общепризнанных элементов



Использование ядра



Заключение

Software Engineering Method and Theory



www.semat.org

В сентябре 2009 года образована Тройка:

Ivar Jacobson – Bertrand Meyer – Richard Soley

«Сегодня некоторые области программной инженерии страдают от незрелых практик. Характерные проблемы включают:

- Распространенность увлечений, более характерная для индустрии моды, нежели для инженерных дисциплин;
- Отсутствие основательной, широко признанной теоретической базы;
- Огромное число методов и их вариантов, разница между которыми плохо понимается и искусственно преувеличивается;
- Отсутствие заслуживающих доверия экспериментальных оценок и подтверждений;
- Разрыв между научными исследованиями и практическим применением»

SEMAT поддерживает процесс переопределения программной инженерии на основе убедительной теории, проверенных принципов и лучших практик, которые:

- включают ядро, состоящее из общепринятых элементов, с возможностью расширения под конкретные нужды;
- затрагивают как вопросы технологии, так и человеческий фактор;
- поддержаны индустрией, наукой, исследователями и пользователями;
- поддерживают расширение, вызванное изменениями в требованиях и технологиях.

Основное направление

Мы поддерживаем процесс **воссоздания программной инженерии**, на основе **убедительной теории**, **проверенных принципов** и **лучших практик**

Последующие шаги

Определение
убедительной
теоретической
базы

Поиск ядра,
состоящего из
широко
признанных
элементов

- Pekka Abrahamsson
- Scott Ambler
- Victor Basili
- Jean Bézivin
- Robert V. Binder
- Dines Bjorner
- Barry Boehm
- Alan W. Brown
- Larry Constantine
- Steve Cook
- Bill Curtis
- Donald Firesmith
- Erich Gamma
- Carlo Ghezzi
- Tom Gilb
- Ellen Gottesdiener
- Sam Guckenheimer
- Robert Glass
- David Harel
- Brian Henderson-Sellers
- Martin Griss
- Capers Jones
- Ivar Jacobson
- Philippe Kruchten
- Harold Lawson
- Robert Martin
- Bertrand Meyer
- James Odell
- Meilir Page-Jones
- Dieter Rombach
- Ken Schwaber
- Alec Sharp
- Richard Soley
- Andrey Terekhov
- Fuqing Yang
- Ed Yourdon

- ABB, Switzerland
- Ericsson, Sweden
- Fujitsu, UK
- Huawei, China
- IBM, USA
- Microsoft, Spain
- Munich RE, Germany
- SAAB, Sweden
- Samsung SDS, Korea
- SEI, USA
- SINTEF, Norway
- Software Engineering Center, Korea
- Swedish Institute of Computer Science, Sweden
- Telecom Italia, Italy
- City of Toronto, Ontario, Canada
- Wellpoint, USA

- Chalmers, Sweden
- Free University of Bozen, Italy
- Florida Atlantic University, USA
- Fudan University, China
- Joburg Centre for Software Engineering (JCSE) at Wits University
- KAIST, Korea
- Nanjing University, China
- **National University of Columbia at Medellin**
- Peking University, China
- **PCS - Universidade de São Paulo**
- Royal Institute of Technology, Sweden
- Software Engineering Institute of Beihang University, China
- Shanghai University, China
- Tsinghua University, China
- Wuhan University, China

- Китай
- Латинская Америка
- Южная Африка
- Россия
- И многие в пути...

Более 1900 участников



Состояние индустрии



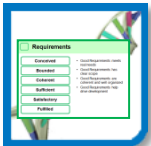
Специфические проблемы



Направления деятельности

- определение убедительной теоретической базы

- поиск ядра, состоящего из общепризнанных элементов



Использование ядра



Заключение

- Уравнения Максвелла
- Теория Большого Взрыва
- Теория Клетки
- Теория Спроса и Предложения
- Общая Теория Преступности
- Общая Теория Меры
- Общая Теория Обработки Информации Человеком
- Теория Организационных Структур

“Нет ничего более практичного, чем хорошая теория” – Курт Левин, 1951

Программная инженерия не испытывает недостатка в теориях



В Программной инженерии есть множество теорий:

- **Конструктивная модель стоимости** Б.Бозма –COCOMO (The Constructive Cost Model)
- **Принцип сокрытия информации** Д.Парнаса,
- **Сцепление и Связность** Л.Константайна,
- **Закон Д.Х.Конвея** "Каждая система структурно подобна коллективу, ее разработавшему»,
- **Теория когнитивных ограничений** Э.В.Дейкстры ("Оператор GOTO считается опасным "),
- **Пошаговая детализация** Н.Вирта,
- **Проектирование по контракту** Б.Мейера,
- и т.д.

Но ни одна из этих теорий не является основополагающей

Программная инженерия не испытывает недостатка в теориях



Программная инженерия не испытывает недостатка в теориях



Базовая теория – Первооснова

SEMAT хочет предоставить базовую теорию, на которую смогли бы опереться более специализированные теории



Состояние индустрии

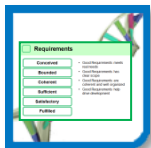


Специфические проблемы



Направления деятельности

- определение убедительной теоретической базы
- ***поиск ядра, состоящего из общепризнанных элементов***



Использование ядра



Заключение

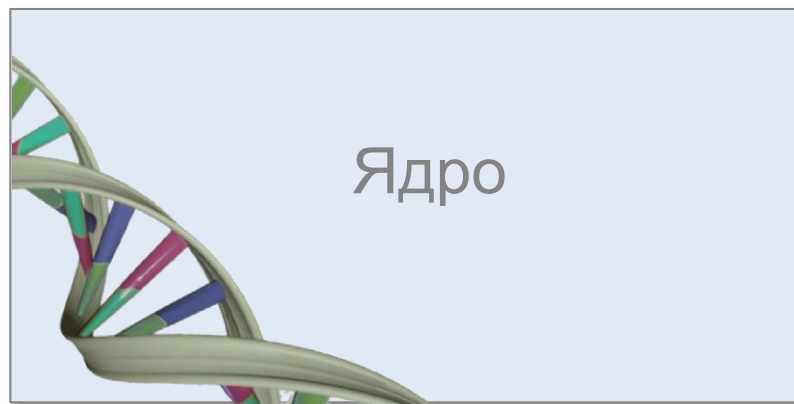
Ядро наиболее распространенных элементов



Ядро включает элементы, которые **универсальны** для всех работ по разработке программного обеспечения

Ядро должно быть выделено из большого количества методов

Ядро – независимо от практики и метода



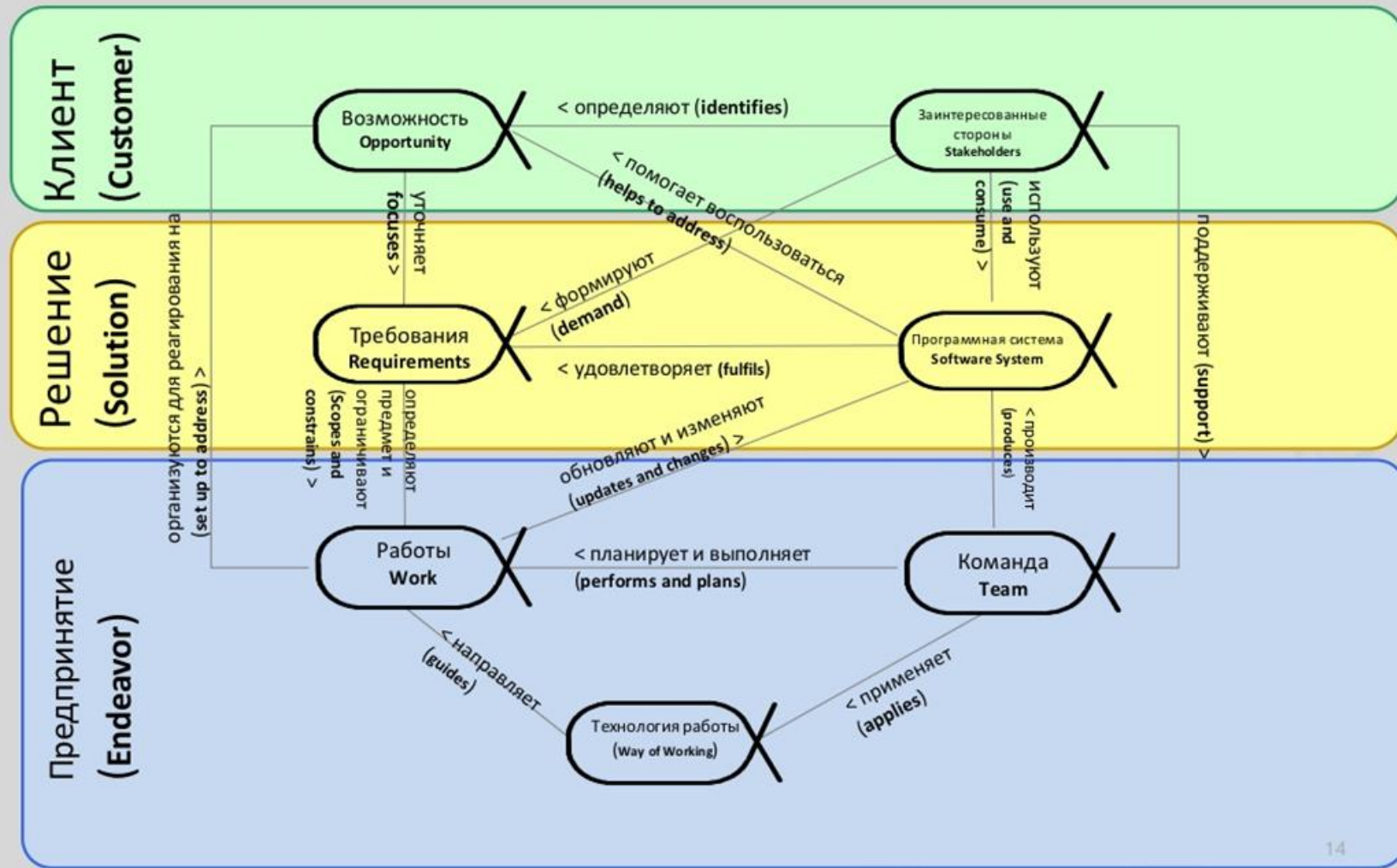
Ядро является сущностью
программной инженерии

«Я освобождаю статую от мрамора»
Приписывается Микеланджело

Перефразируя Микеланджело :
«Мы освобождаем ядро от методов»



- ***ядро исполнимо*** - способ, при помощи которого оно управляется с «тем, с чем мы работаем», альфа (alpha), **обязательный элемент деятельности**, используемый **для оценки прогресса** и его «здоровья», т.е. **продуктивности состояния**
- ***ядро расширяемо*** – содержит способ, которым оно может быть расширено для поддержки разных проектов
- ***ядро практично*** - является ясным, осознанным способом мышления, сфокусированным на поддержке профессионалов в области разработки ПО в том, **как** они выполняют свою работу.







Клиент
(Customer)



Исследовать возможности
Explore Possibilities



Понять потребности
заинтересованных сторон
Understand Stakeholder Needs



Обеспечить удовлетворение
заинтересованных сторон
Ensure Stakeholder Satisfaction



Использовать систему
Use the System

Решение
(Solution)



Понять требования
Understand the Requirements



Спроектировать
систему
Shape the System



Изготовить систему
Implement the System



Протестировать
систему
Test the System



Развернуть систему
Deploy the System



Управлять
системой
Operate the System

Предприятие
(Endeavor)



Подготовиться к работе
Prepare to do the Work



Координировать дела
Coordinate Activity



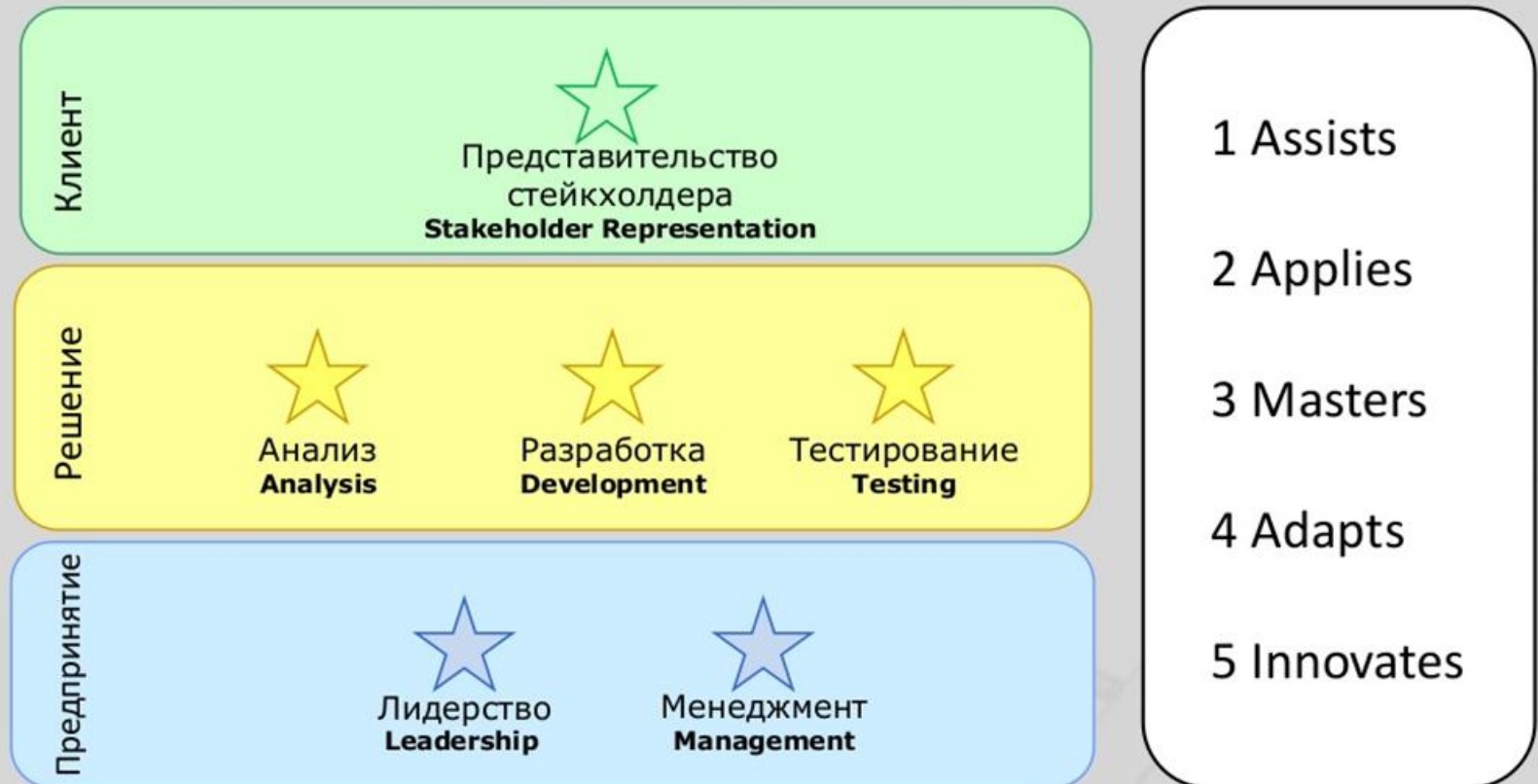
Поддерживать команду
Support the Team



Отслеживать прогресс
Track Progress



Завершить работу
Stop the Work





Состояние индустрии

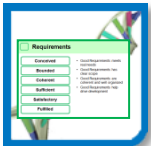


Специфические проблемы



Направления деятельности

- определение убедительной теоретической базы
- поиск ядра, состоящего из общепризнанных элементов

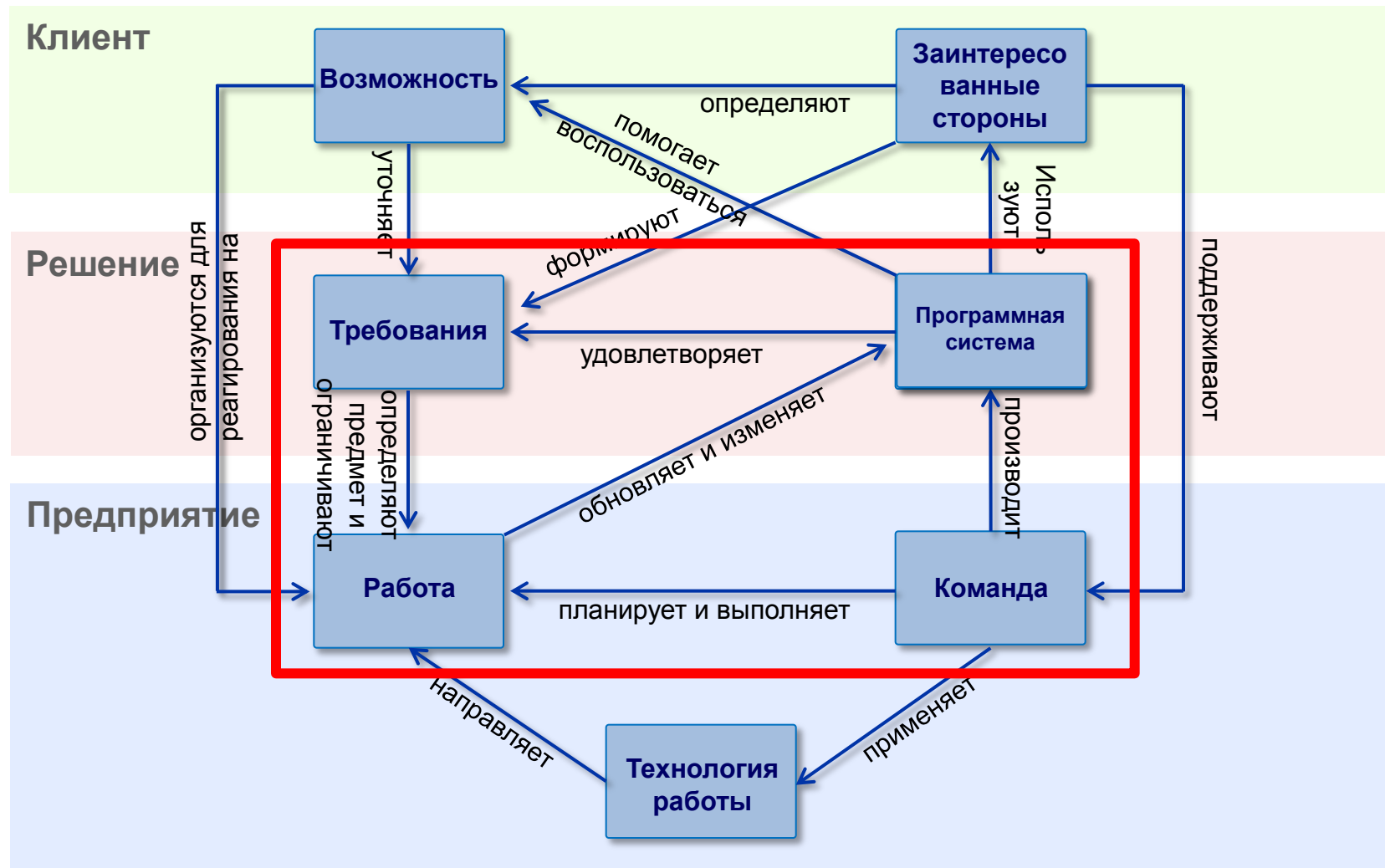


Использование ядра

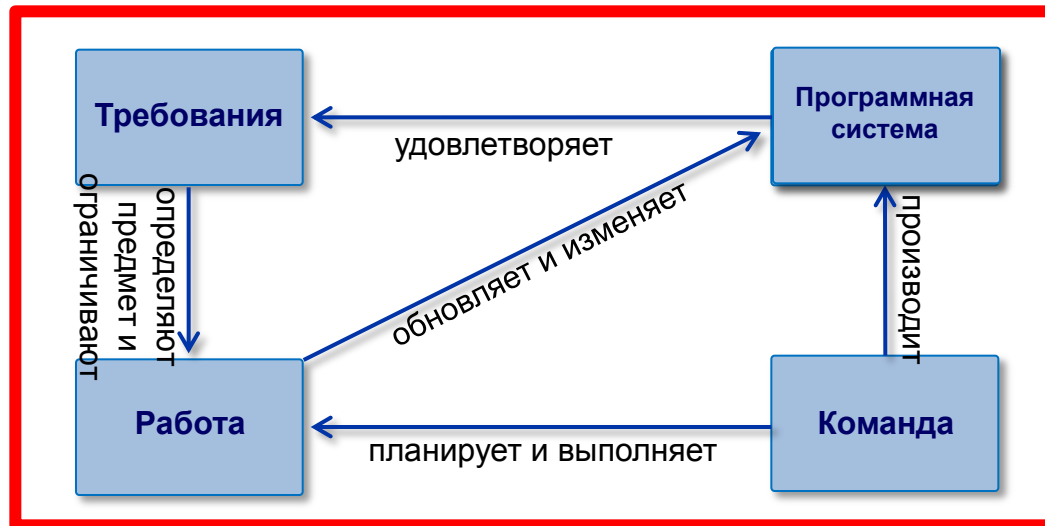


Заключение

Пример: выборка из Ядра



Пример: выборка из Ядра



План : Определение текущего состояния



Требования

Требования	Требования	Требования	Требования	Требования	Требования
Задуманы	Ограничены	Согласованы	Достаточны	Приемлемы	Удовлетворены
Создание системы согласовано с начальным составом заинтересованных сторон. Определены пользователи и заказчики. Есть ясно понимаемая выгода от новой системы.	Назначение, критерии успешности системы ясны. Выбран механизм для управления требованиями. Учитываются ограничения и допущения.	Описанные требования дают ясную картину системы. Противоречивые требования выявлены. Объяснены важные сценарии использования. Приоритеты требований ясны.	Требования адекватно описывают решение и приемлемы для заинтересованных сторон. Уровень изменений по отношению к согласованным требованиям низок и находится под контролем.	Системные требования следует внедрить. Достаточно требований имплементировано.	Системные требования внедрены, как полностью удовлетворяющие. Никакой нереализованный пункт требований не мешает принятию системы. Заинтересованные стороны принимают требования как правильные.
1/6	2/6	3/6	4/6	5/6	6/6

Программная система

Программная система	Программная система	Программная система	Программная система	Программная система	Программная система
Архитектура выбрана	Демонстрируемая	Подходит для использования	Готовая	Используемая	Выведена из эксплуатации
Архитектура выбрана, чтобы выявить ключевые технические риски. Согласованы критерии для выбора архитектуры. Выбраны платформы, технологии, языки. Покупайте, стройте, повторно используйте принятые решения.	Исполнимая версия системы демонстрирует, что архитектура подходит для заданных целей. Поддерживает функциональное и нефункциональное тестирование. Осуществляется критический интерфейс и конфигурация системные.	Система пригодна к употреблению и имеет требуемые качественные характеристики. Система может использоваться пользователями. Функциональность и производительность протестированы и приняты. Количество дефектов приемлемо. Контент релиза известен.	Система (в целом) принята для внедрения. Спонсоры, пользователи, заинтересованные стороны приняли систему, как пригодную для своих целей. Установка и документация доступны. Имеется эксплуатационная поддержка.	Система эксплуатируется в промышленной среде. Система доступна для предполагаемых пользователей. По крайней мере один экземпляр системы полностью работоспособен. Система поддерживается согласованным уровнем обслуживания.	Система больше не поддерживается. Обновления в систему больше не вносятся. Система была заменена или остановлена.
1/6	2/6	3/6	4/6	5/6	6/6

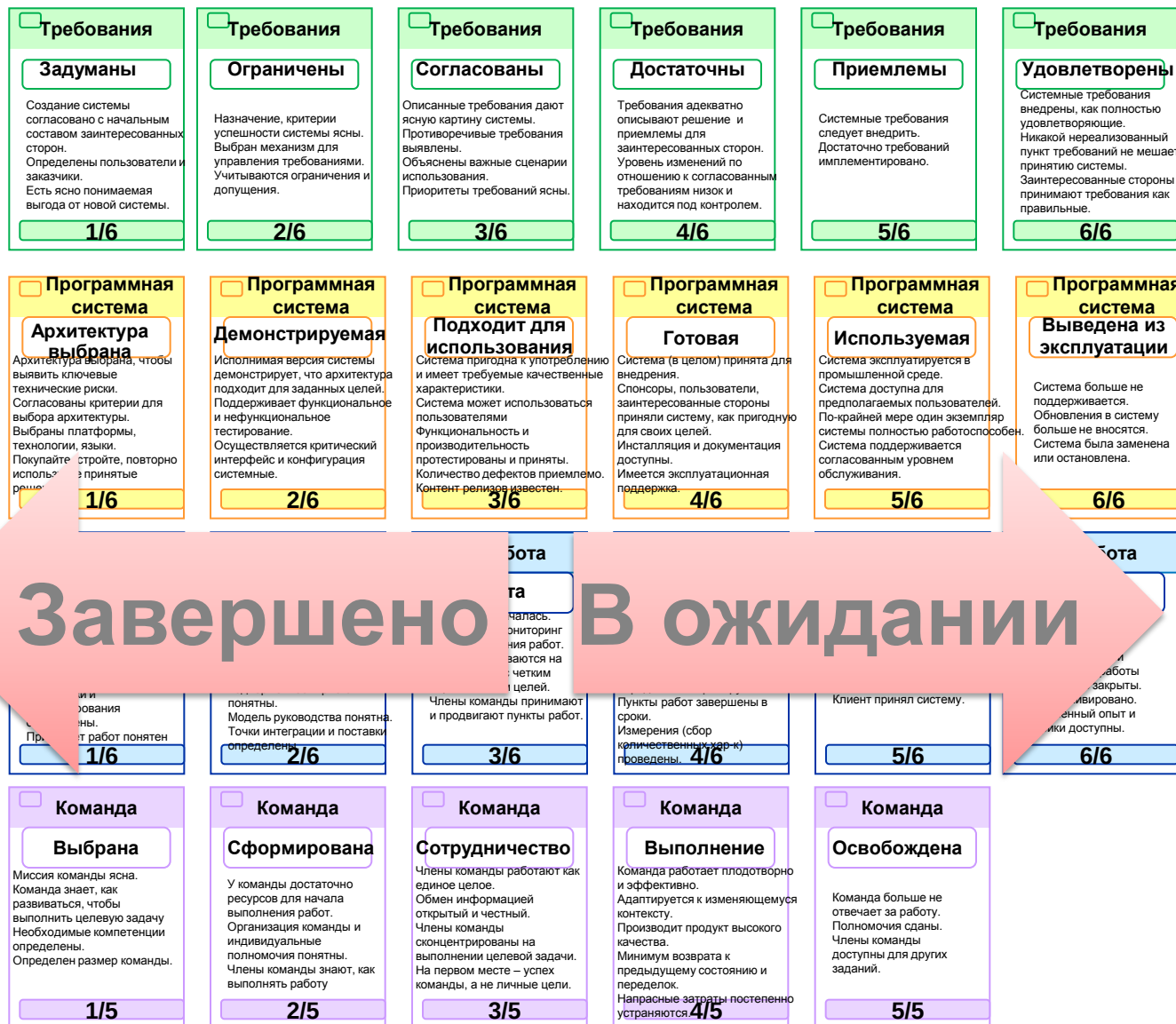
Работа

Работа	Работа	Работа	Работа	Работа	Работа
Инициирована	Подготовлена	Начата	Контролируется	Заканчивается	Закрыта
Инициатор работ и клиент известны. Цель работ и ограничения ясны. Модель спонсорской поддержки и финансирования определены. Приоритет работ понятен.	Затраты и объем работ понятны. Финансирование выполняется. Доступность ресурсов и подверженность рискам понятны. Модель руководства понятна. Точки интеграции и поставки определены.	Разработка началась. Проводится мониторинг хода выполнения работ. Работы разбиваются на шаги (этапы) с четким определением целей. Члены команды принимают и продвигают пункты работ.	Работы продвигаются нормально, риски управления уровни продуктивности приемлемы. Незапланированные работы и переделки контролируются. Пункты работ завершены в сроки. Измерения (сбор количественных данных) проведены.	Работы доведены до результатов. Ожидаемые результаты работ достигнуты. Клиент принял систему.	Все оставшиеся служебные задачи завершены, и работы официально закрыты. Все заархивировано. Накопленный опыт и метрики доступны.
1/6	2/6	3/6	4/6	5/6	6/6

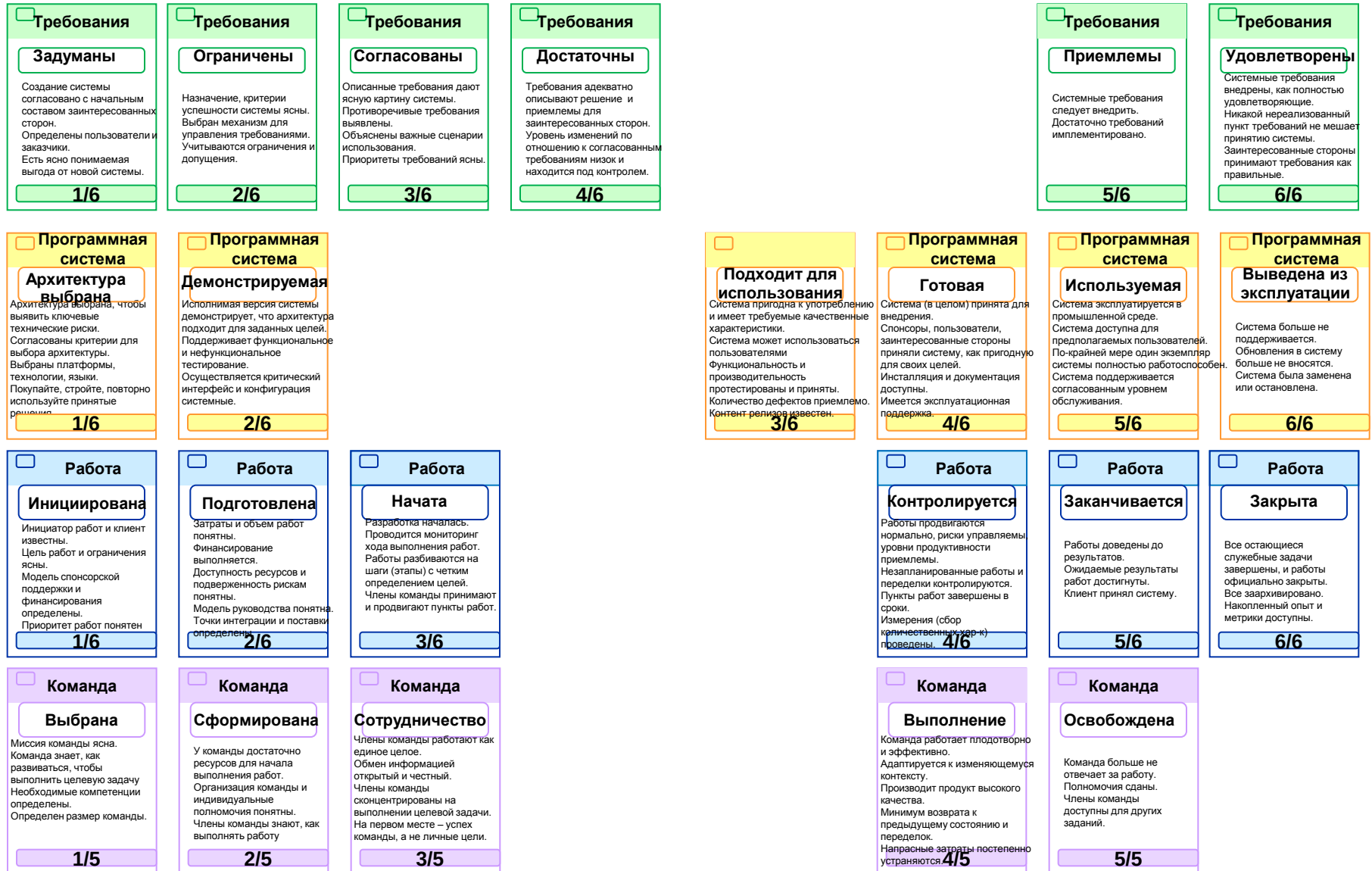
Команда

Команда	Команда	Команда	Команда	Команда
Выбрана	Сформирована	Сотрудничество	Выполнение	Освобождена
Миссия команды ясна. Команда знает, как развиваться, чтобы выполнить целевую задачу. Необходимые компетенции определены. Определен размер команды.	У команды достаточно ресурсов для начала выполнения работ. Организация команды и индивидуальные полномочия понятны. Члены команды знают, как выполнять работу	Члены команды работают как единое целое. Обмен информацией открытый и честный. Члены команды сконцентрированы на выполнении целевой задачи. На первом месте – успех команды, а не личные цели.	Команда работает плодотворно и эффективно. Адаптируется к изменяющемуся контексту. Производит продукт высокого качества. Минимум возврата к предыдущему состоянию и переделок. Напрасные затраты постепенно устраняются.	Команда больше не отвечает за работу. Полномочия сданы. Члены команды доступны для других заданий.
1/5	2/5	3/5	4/5	5/5

План : Определение текущего состояния



План : Определение последующего состояния



План: как достигнуть следующего состояния



☐ Требования

Приемлемы

- Системные требования следует внедрить
- Достаточно требований имплементировано

5/6

☐ Программная система

Подходит для использования

- Система пригодна к использованию и имеет требуемые качественные характеристики
- Система может использоваться пользователями
- Функциональность и производительность протестированы и подтверждены
- Количество дефектов приемлемо
- Контент релизов известен

3/6

☐ Работа

Контролируется

- Работа продвигается нормально, риски управляемы, уровни продуктивности приемлемы
- Незапланированные работы и переделки контролируются
- Пункты работы завершены в сроки
- Измерения (сбор количественных характеристик) проведены

4/6

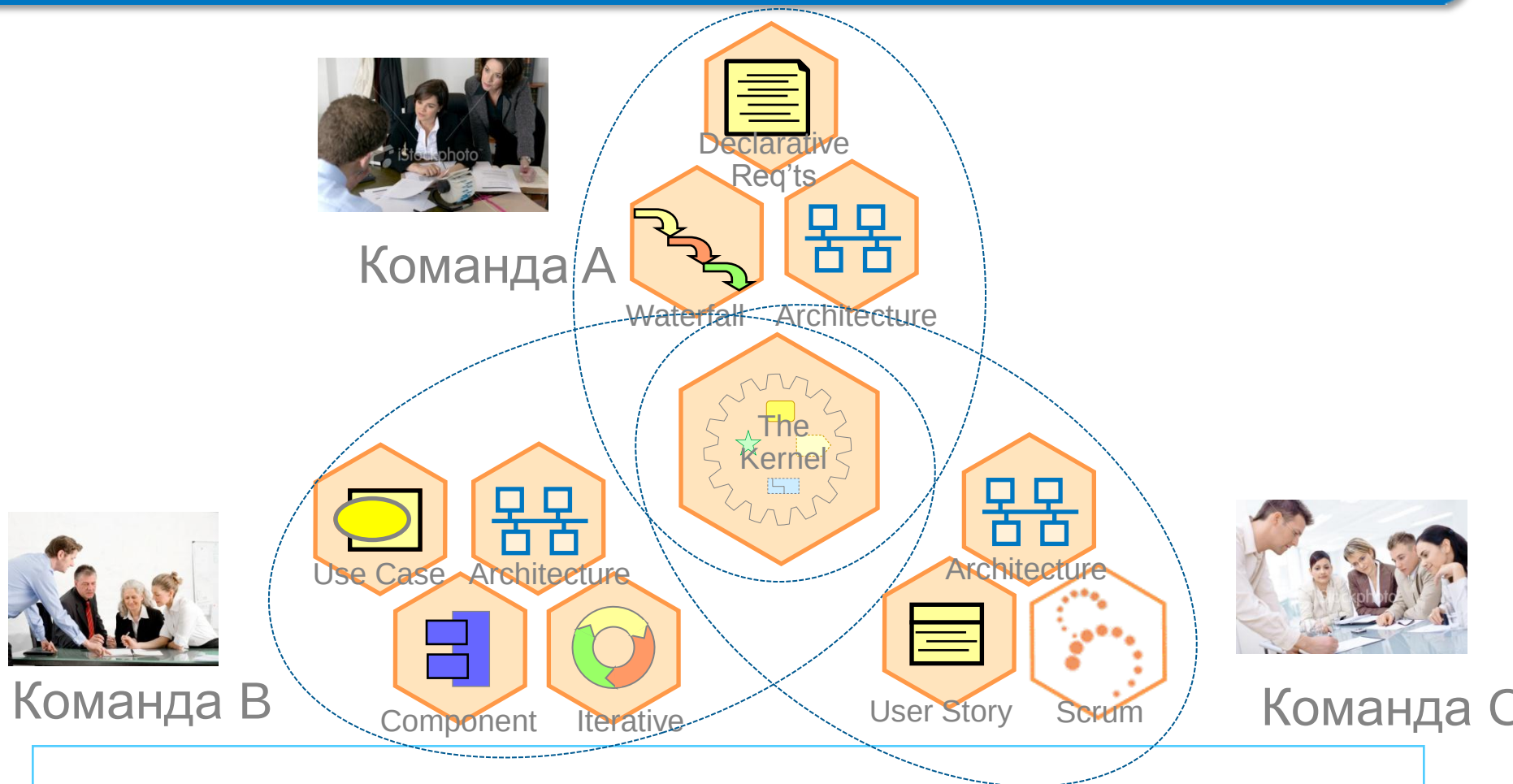
☐ Команда

Выполнение

- Команда работает плодотворно и эффективно
- Адаптируется к изменяющемуся контексту
- Производит продукт высокого качества
- Минимум возврата к предыдущему состоянию и переделкам
- Напрасные затраты постепенно устраняются

4/5

Ядро обеспечивает совместно используемую систему отсчета



На базе одного и того же ядра может быть построено много методов

Важно, что ядро сфокусировано на то, чтобы сделать проект **ВЫПОЛНИМЫМ...**



Требования	
Задуманы	•Хорошие требования соответствуют реальным потребностям
Ограничены	
Согласованы	•Хорошие требования имеют четкую область действия
Достаточны	•Хорошие требования последовательны и правильно организованы
Приемлемы	
Удовлетворены	•Хорошие требования помогают проводить разработку

Требования	
Задуманы	
Создание системы согласовано с начальным составом заинтересованных сторон. Определены пользователи и заказчики. Есть ясно понимаемая выгода от новой системы.	
1 / 6	

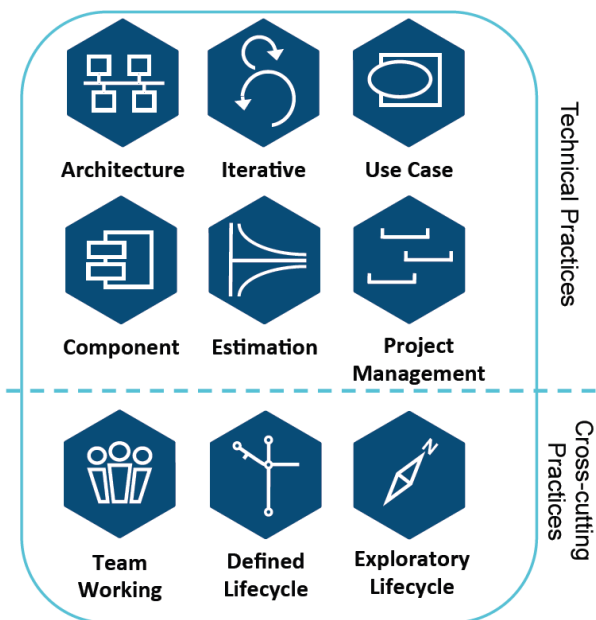
Каждое состояние определяет чеклист

... и помочь Вам понять прогресс, задачи и "здоровье" проекта

Реинжиниринг вашего программного процесса: Rule Financial

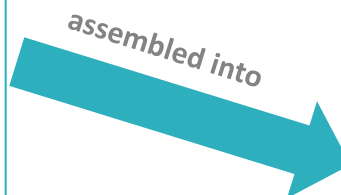


The Sector Specialists



Technical Practices

Cross-cutting Practices



Defined Method

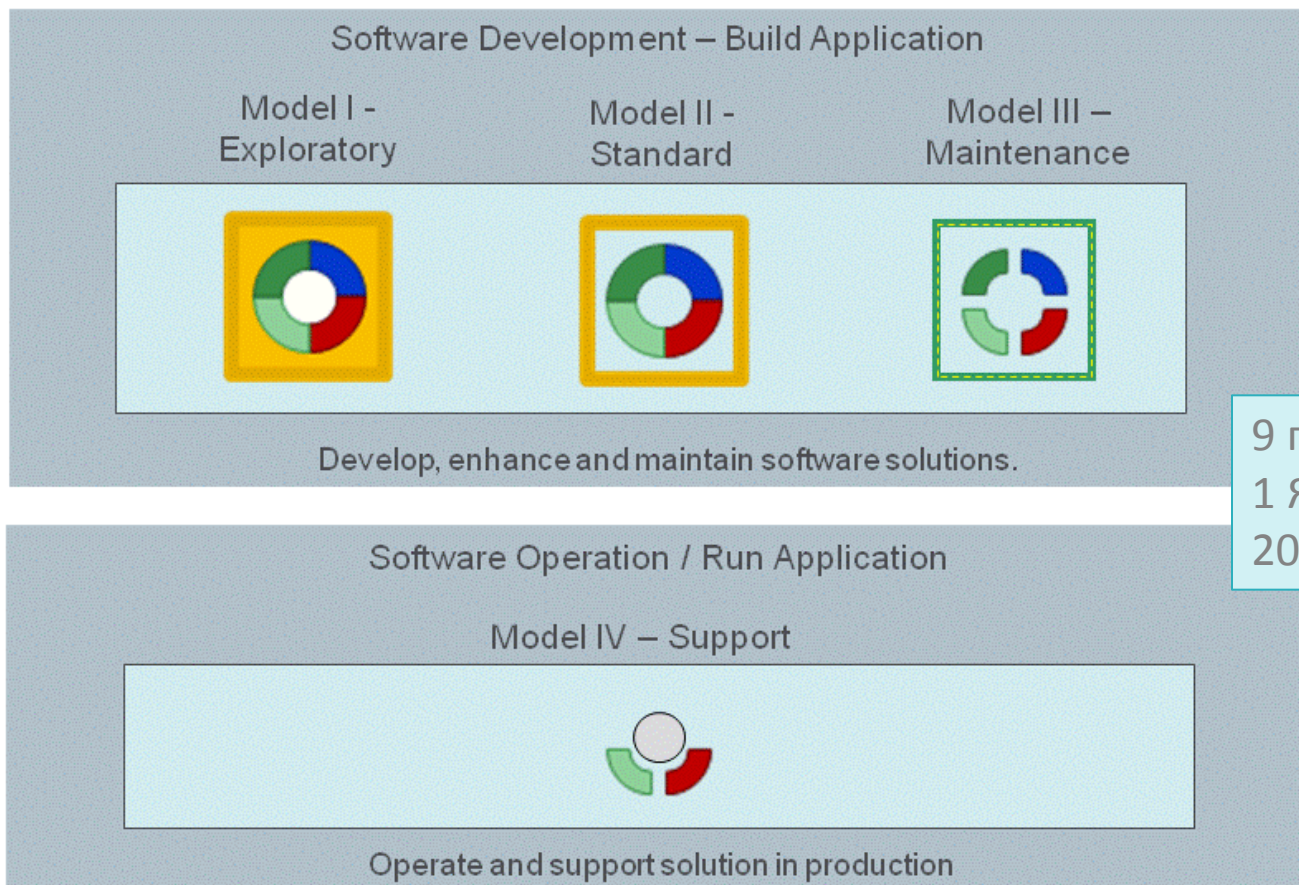


Exploratory Method

Проект: rulebook™

-это гибкий набор экономных и динамичных практик, приводящих к упрощенным процессам, настроенным так, чтобы соответствовать потребностям (см. www.rulefinancial.com).

Реинжиниринг вашего программного процесса: MunichRE



9 принципов
1 Ядро
20 Практик



Состояние индустрии

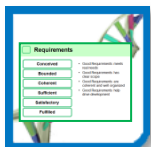


Специфические проблемы



Направления деятельности

- определение убедительной теоретической базы
- поиск ядра, состоящего из общепризнанных элементов



Использование ядра



Заключение



Новые ключевые принципы

- Будьте динамичны, работая с методами



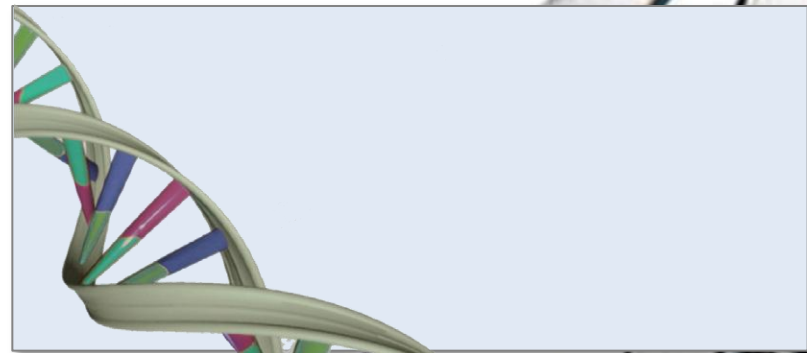
- Будьте динамичны, работая с методами
- **Разделяйте Проблемы,**
например, четко разделяйте, что требует
малоопытный разработчик, и что требуют
опытные разработчики..



Этот подход является практическим

- Ядро является общей основой

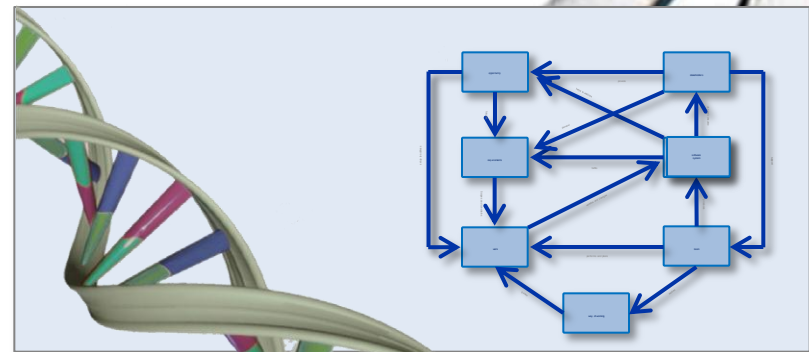
Ядро



- Ядро является общей основой
- Внутри ядра находятся элементы. Они имеют характеристики для измерения прогресса и состояния.
- Сосредоточьтесь на применении метода, а не на описании метода



Ядро



- Ядро является общей основой
- **Практики располагаются над ядром**
- Практики: Use Cases, Test-Driven Development, и т.д.

Практики



Этот подход является практическим

- Ядро является общей основой
- Практики располагаются над ядром
- **Метод – это набор практик**

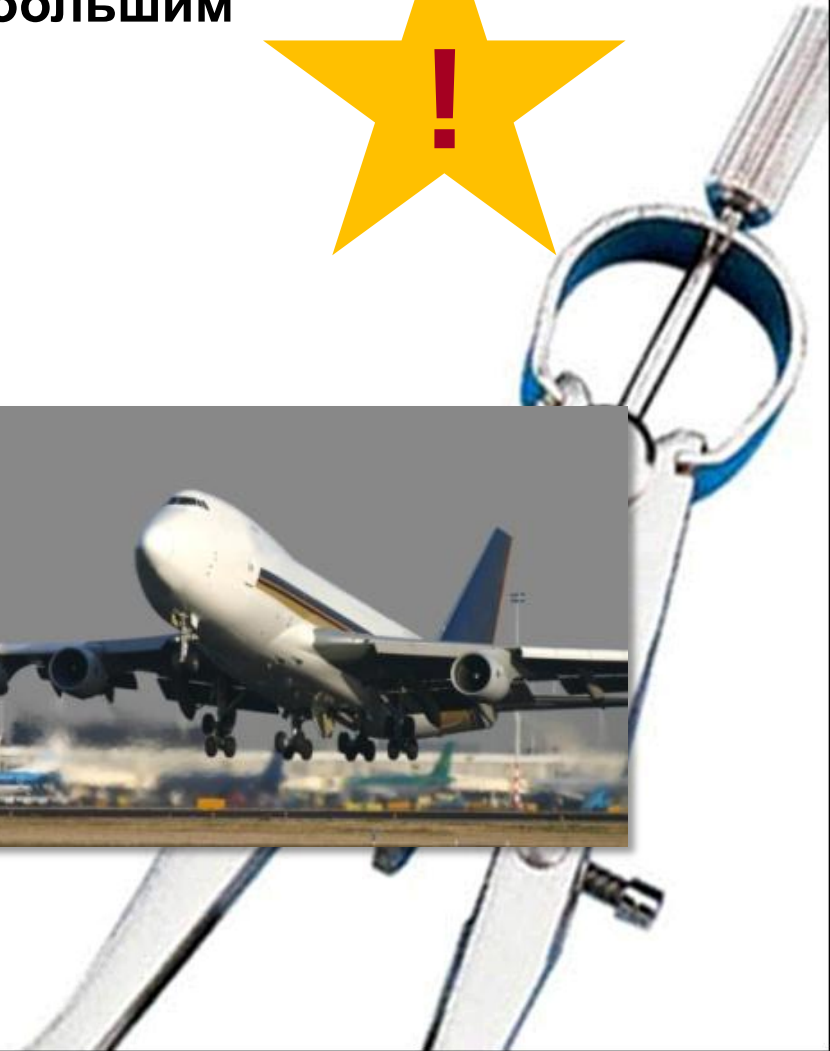
Метод



Этот подход является масштабируемым



От очень маленьких проектов к большим
От одного продукта ко многим
От одного метода ко многим



В двух словах – что БУДЕМ менять



Производство

Научная сфера

Профессионалы

Хотят стать экспертами. Задачи :
- **навыки и опыт нелегко снова использовать для нового продукта.**
- карьерный путь проходит по извилистой дороге от одной рекламной шумихи к другой.

Руководители

В больших компаниях существует много процессов.
Задачи :
- **Повторное использование практики**
- Повторное обучение
- “Повторное использование” людей
- **Эволюционные улучшения трудны**

Образование

Обучение конкретным методологиям, таким как Scrum, RUP, вместо освоения универсальной методологической базы;
универсалы не появляются

Исследования

Разрыв между исследованием и производством
Нет общепризнанной теории

SEMAT ориентирует все программистское сообщество

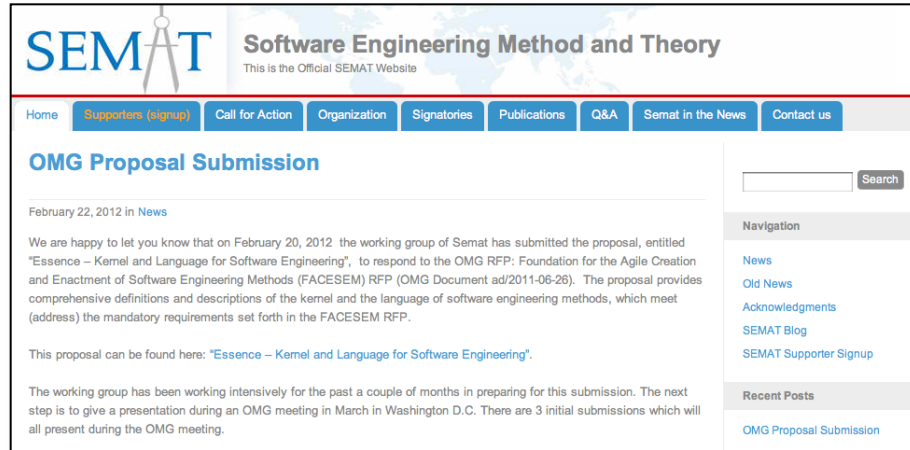
«Похоже, что конференция SEMAT в Цюрихе (2010) будет историческим событием, таким как сессия НАТО в Гармише»

Уоттс Хамфри (СММИ)

В 1968 году на знаменательной конференции НАТО «Программная Инженерия» в Гармише (Германия) впервые появился термин «программная инженерия»



Что вы можете сделать сейчас



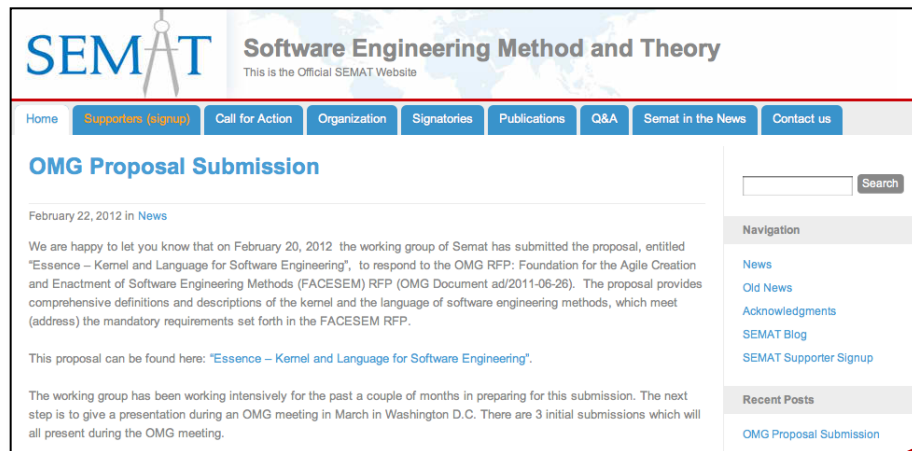
Становитесь
сторонником

www.semat.org

Присоединяйтесь к
SEMAT

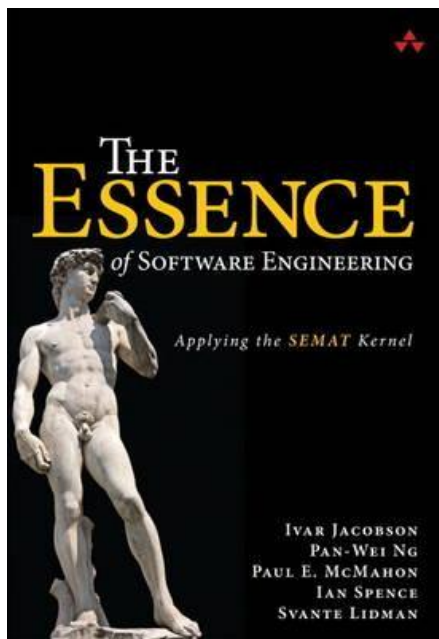


Что вы можете сделать сейчас



Становитесь сторонником
www.semat.org
Присоединяйтесь к SEMAT

Читайте новую книгу на
http://www.amazon.com/The-Essence-Software-Engineering-Applying/product-reviews/0321885953/ref=dp_top_cm_cr_acr_txt?ie=UTF8&showViewpoints=1

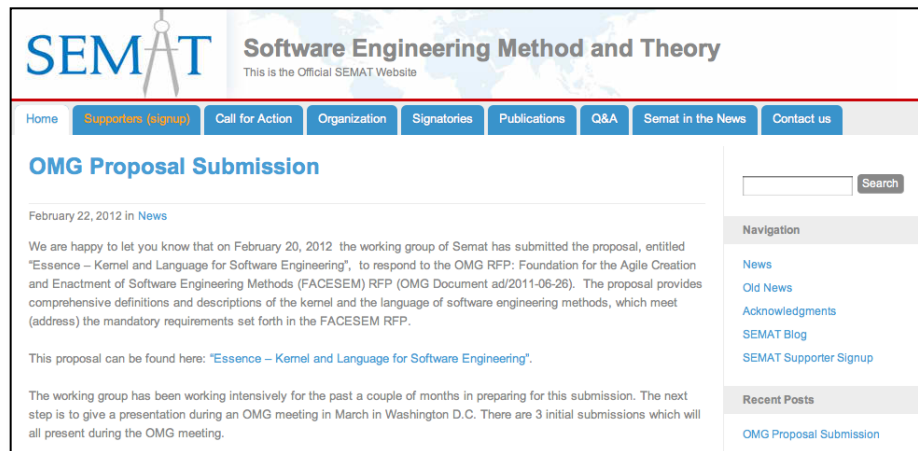


«Essence – Kernel and Language for
Software Engineering Methods»

<http://semat.org/wp-content/uploads/2012/02/12-08-15.pdf>

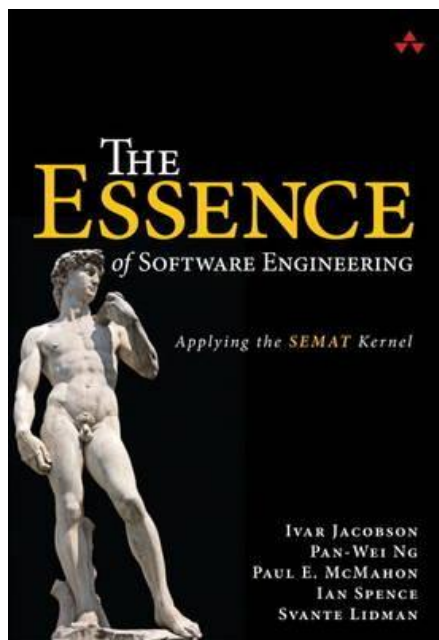
<http://semat.org/wp-content/uploads/2012/02/12-08-15.pdf>

Что вы можете сделать сейчас



Становитесь сторонником
www.semat.org
Присоединяйтесь к SEMAT

Читайте новую книгу на
http://www.amazon.com/The-Essence-Software-Engineering-Aplying/product-reviews/0321885953/ref=dp_top_cm_cr_acr_txt?ie=UTF8&showViewpoints=1



Присоединяйтесь
к нам!!

<http://semat.org/wp-content/uploads/2012/02/12-08-15.pdf>

1. Ivar Jacobson and Bertrand Meyer: “Methods need theory” Dr. Dobb's Journal, August 06, 2009. Online at <http://www.drdobbs.com/architecture-and-design/219100242>
2. Ivar Jacobson and Ian Spence: “Why we need a theory for software engineering” Dr. Dobb's Journal, October 02, 2009. Online at <http://www.drdobbs.com/architecture-and-design/220300840>
3. Ivar Jacobson, Bertrand Meyer, and Richard Soley: “Call for Action: The Semat Initiative” Dr. Dobb's Journal December 10, 2009. Online at <http://www.drdobbs.com/architecture-and-design/222001342>
4. Ivar Jacobson, Bertrand Meyer, and Richard Soley: “The Semat Vision Statement” online at <http://blog.paluno.uni-due.de/semat.org/wp-content/uploads/2012/03/SEMAT-vision.pdf>
5. Ivar Jacobson, Shihong Huang, Mira Kajko-Mattsson, Paul McMahon, Ed Seymour. “Semat - Three Year Vision” Programming and Computer Software 38(1): 1-12 (2012), Springer 2012. DOI: 10.1134/S0361768812010021.
6. “Essence – Kernel and Language for Software Engineering”(OMG Document number: ad/2011-02-04) A proposal submitted to OMG RFP (OMG Document ad/2011-06-26)
7. Jacobson I., Ng P. W., McMahon P., Spence I., Lidman S. 2013. *The Essence of Software Engineering – Applying the SEMAT Kernel*. Addison-Wesley. Forthcoming in January, 2013, but available now online in a pre-publication version on safaribooksonline.com
8. “Re-founding Software Engineering – SEMAT at the Age of Three”, Ivar Jacobson, Ian Spence, Pontus Johnsson, Mira Kajko-Mattsson, ASE 2012.

полный список литературы по данной теме см. на www.semat.org





SEMAT Russian Chapter

- OMG website
- www.omg.org
- SEMAT website
- www.semat.org
- SEMAT@ec-leasing.ru

<http://semat.org/wp-content/uploads/2012/02/12-08-15.pdf>



Спасибо за внимание!

