

УДК 681.3.06

## SEMAT – ВЗГЛЯД НА ТРИ ГОДА ВПЕРЕД

© 2012 г. Джекобсон А., Хуан Ш., Кайко-Мэтсон М., МакМэхон П., Сеймур Э.  
 E-mail: *ivar@ivarjacobson.com, shihong@fau.edu, mekm2@kth.se, petmahon@actm.org,*  
*Ed.Seymour@uk.fujitsu.com*

*В работе над статьей принимали участие: экспертная комиссия Semat (Бертран Мейер, Ричард Соли), Арн Бэр, Дональд Фаерсмит, Кэперс Джонс, и Гарольд „Бад“ Лоусон*

Поступила в редакцию 09.10.2011 г.

Эта статья преследует три цели: в ней, во-первых, резюмируются текущие результаты проекта Semat; во-вторых, изложены направления его развития для тех, кто активно работает в сообществе, связанном с этим проектом; в-третьих, предоставлен базовый материал для поиска финансирования от таких организаций, как Европейское Сообщество и т.п. Финансовая поддержка необходима, чтобы поддержать продолжение Semat и его преобразование в более широкую деятельность сообщества, так как большинство людей работают в проекте безвозмездно. Статья может оказаться как слишком длинным, так и слишком коротким изложением сути проекта для широкой аудитории. Однако мы намерены сделать нашу работу полностью прозрачной, и поэтому публикуем все ее результаты. Мы хотели бы получить реакцию и комментарии от сочувствующих проекту, чтобы учесть их в планах. Поэтому, чтобы лучше удовлетворить особые потребности практики, бизнеса и исследователей, мы одновременно работаем над несколькими близкими статьями.

## 1. ВВЕДЕНИЕ

В конце 2009 года Айвар Джекобсон, Бертран Мейер и Ричард Соли выдвинули инициативу, названную Semat (Software Engineering Method and Theory, метод и теория программной инженерии), с целью переосмысления основ программной инженерии как строгой дисциплины. Они признали, что естественная тенденция в этой области – вносить минимальные изменения в системы для постепенного приближения к их правильности, но этим путем нельзя следовать, если мы хотим поддержать конкурентоспособность бизнеса информационных технологий (ИТ) и помочь ему соответствовать требованиям общества. Они стали утверждать необходимость переосмысления программной инженерии на прочной основе, используя все теоретические и практические результаты, полученные за последние пять десятилетий.

Цель этой статьи – показать, насколько далеко продвинулась реализация этих идей за первый год, и обрисовать ожидаемые через три года

результаты, т.е., ситуацию начала 2014 года. Детальный план проекта будет представлен в отдельных документах.

## 2. „ВЕЛИКАЯ МЕЧТА“

Исходный призыв к действиям [3], составленный в сентябре 2009, указал основные вопросы и проблемы программной инженерии, такие как подверженность моде и увлечениям, отсутствие теоретической основы, избыток уникальных методов, которые трудно сравнить, дефицит экспериментальных оценок и проверок, а также разрыв между исследованиями и их практическим использованием в промышленной разработке (см. Приложение 1).

Для преодоления этого было предложено решение – „Великая мечта“ (Grand Vision [4]), в рамках которой программная инженерия должна быть переосмыслена на базе убедительной теории, проверенных принципов и лучших практик. Основу дисциплины должны составить общепринятые элементы, выделяемые в *ядро*. Ядро должно стать фун-

даментальной основой, помогающей практикам (например, архитекторам, проектировщикам, разработчикам, тестировщикам, инженерам по требованиям, инженерам-технологам, менеджерам и т. д.) сопоставлять различные методы и принимать лучшие решения в своих проектах.

Для воплощения „Великой мечты“ нужна многолетняя работа, определенно более длительная, чем три года. Эта работа стартовала в начале 2010. Первый ее этап [4] описывается в Разделе II. В Разделе III обсуждаются текущие итоги Semat, и дан набросок дальнейшего пути к выработке фундаментальных и практически значимых результатов для всех людей, имеющих дело с программным обеспечением (ПО).

### 3. ПЕРВЫЙ ЭТАП – СОЗДАНИЕ ОСНОВЫ ДЛЯ SEMAT

Чтобы выстроить заново программную инженерию, нужно сделать то, что никогда не делалось прежде – выделить „ядро общепринятых элементов“, способное стать фундаментальным, общим основанием этой дисциплины.

#### 3.1. *Общее основание программной инженерии*

Разработчики ПО создают его уже в течение более 50 лет. Независимо от производимого кода, создаваемых систем, вырабатываемых решений, применяемых методов или вовлеченных организаций, общее основание – *ядро* из элементов, представленных повсеместно используемыми понятиями и характеристиками – присутствует в рамках любых программных проектах. Примерами таких важных понятий являются работа (или задание), команда, требование, программная система, бизнес-возможность и заинтересованное лицо. Это ядро представляет собой самую суть программной инженерии.

Определение этого ядра предоставит программистам-практикам инструменты, необходимые, чтобы лучше понимать, компоновать и сравнивать отдельные практики и методы, и делать свою работу более эффективно. На его основе компании смогут строить целостную и просчитываемую административную структуру, в то же время позволяющую разработчикам

свободно использовать любимые техники. Оно определит развитие образования, помогая формировать новые учебные планы и устанавливать цели личного развития, и поддержит исследования за счет предоставления для них контекста и общепонятных предметов, имеющих практическое значение. Более того, все это уменьшит влияние моды и увлечений, распространенных сегодня в индустрии, и возвестит эру большего прагматизма и объективности.

Поиск составляющих такого ядра крайне важен. Выделяя универсальные и значимые элементы, мы руководствуемся максимой „Вы достигли совершенства не тогда, когда нечего добавить, а когда нечего убрать“.<sup>1</sup> Мы также стараемся согласовать со многими заинтересованными лицами выделяемые составляющие, которые „нельзя убрать“.

Таким образом, общепринятое ядро является сущностью программной инженерии. При использовании термина „общее основание“ в этой статье имеется в виду достигнутое согласие в понимании этой сущности. Под „ядром“ подразумевается реализация общего основания – общее основание является спецификацией, а ядро – ее реализацией.

#### 3.2. *Использование разделения ответственности в Semat*

Ключевой принцип Semat состоит в том, чтобы использовать все имеющиеся отношение к делу наработки программной инженерии и не исключать чего-либо, что является или может быть значимым для какой-либо группы интересов. Например, главная наша целевая группа – это программисты-практики, но специальное расширение общего ядра предназначено для инженеров-технологов. Несмотря на то, что основной предметной областью является программная инженерия, рассматриваются также вопросы системной инженерии; хотя мы сосредотачиваемся на базовых вопросах, проект открыт для добавления деталей; наши определения являются общими, но также допускаются их расширение и специализация. Эта неоднозначность достигается следованием

<sup>1</sup> Антуан де Сент-Экзюпери.

принципу разделения ответственности. Он позволяет определить базовые вещи, а затем за счет расширения без изменений или усложнений вносить уточнения, необходимые для конкретных ситуаций (см. Приложение 2).

Важнейшим решением проекта стал выбор программистов-практиков в качестве основной целевой группы. Есть проекты совершенно разных масштабов, и в огромном количестве проектов, выполняемых в малых и средних компаниях, нет инженеров-технологов, отвечающих за процессы разработки. Более того, если практики не воспримут результаты этой инициативы, откровенно говоря, она останется в истории лишь как игра ума.

### 3.3. Требования

#### 1. Получение требований

До сих пор различные методы (здесь мы не различаем такие понятия, как метод, методология и процесс) описывались, в основном, отдельно друг от друга. Каждый метод рассматривался как уникальная сущность, описываемая с помощью своих собственных языка и набора понятий, без использования какого-либо общепринятого понятийного ядра. Выделение такого ядра позволит методологам описывать новые подходы, не изобретая уже известные и согласованные вещи.

Хотя понятие практики связано с методом, оно отличается от него. Оно часто использовалось в программной инженерии за последние 50 лет. Интуитивное понимание заключается в том, что команда обычно использует один метод, но много практик. Таким образом, метод представляется как нечто большее и более сложное, чем отдельная практика.

Обычно у каждой команды разработки ПО есть свой собственный метод разработки. Поэтому можно считать, что на сегодняшний день существует, вероятно, более 100000 методов, причем многие из них никогда не описывались. Это нормально. Нужно ожидать, что существует огромное число методов, но количество отдельных значимых утилитарных практик должно быть намного меньше. Во всей литературе по программной инженерии было определено всего несколько сотен практик.

Таким образом, возможность создавать новые методы на основе выбора нужных практик, каждая из которых описывается при помощи ядра базовых элементов, является ключевым требованием Semat. В Приложении 3 описано отношение между методом, практиками, ядром и языком.

#### 2. Формулировка цели

Наша цель состоит в том, чтобы создать масштабируемые, расширяемые и удобные язык и ядро понятий, которые позволят описывать суть имеющихся и будущих методов и практик так, чтобы разработчики-практики могли их компоновать, сопоставлять, оценивать, адаптировать, внедрять, использовать, моделировать и измерять, исследователи могли их изучать, а преподаватели – обучать им.

#### 3. Ключевые требования

Немного повторяя уже сказанное, данный раздел определяет ключевые требования, связанные с первой фазой проекта Semat. Они не изменились сколь-нибудь существенно направлениям с тех пор, как были изложены в концептуальном плане первого года в феврале 2010 [4].

Одно из ключевых требований – выделить и описать ядро понятий, включающее наиболее существенные элементы программной инженерии. Тогда это ядро служило бы словарем или картой территории программной инженерии. Такая карта использовалась бы как основа, с помощью которой мы могли бы определить любые существующие или предвидимые в ближайшем будущем методы или практики. Это ядро также должно быть расширяемым, чтобы быть способным включать новые технологии, новые образцы общественного взаимодействия и новые исследования. Заметим, что данное требование – это тоже пример применения принципа разделения ответственности: отделение ядра от особенностей различных методов.

Каждая практика будет описываться с использованием ядра, и мы предполагаем возможность объединять практики вместе, образуя составные практики. Метод можно представить как более высокоуровневую практику, которая может быть конкретизирована и использована. Есть

достаточно доказательств на основе большого количества экспериментов, что несколько таких композиций приводят к какому-то методу, и что каждый существующий метод мог бы быть описан как композиция дополняющих друг друга практик. Ядро позволяет определять методы, многократно используя практики (см. Приложение 3).

Однако, возможность практического определения методов с высоким уровнем переиспользования, это еще не все, что мы хотели получить. Методы, определяемые с помощью ядра, имеют динамику, позволяющую контролировать результаты использования практик, а в итоге (например, в конце итерации или проекта) помогающую ретроспективно оценить метод и адаптировать его на основе полученного опыта. Это меняет традиционное понимание метода.

Обычно описание метода представляется как то, что нужно осуществить, а входящие в это описание деятельности выполняются практиками (аналитиками, разработчиками, тестировщиками, руководителями проектов и т. д.) в некотором порядке, чтобы получить результат, заданный определением. Точка зрения, что „команда – это компьютер, процесс – это программа“, не подходит для такой творческой работы, как создание ПО, которое требует гибкости, интенсивного сотрудничества, и часто основано на методе проб и ошибок.

В итоге ядро определяется с помощью проблемно-ориентированного языка, который обладает статической основой (синтаксис и правила корректности), позволяющей эффективно определять методы, и дополнительными динамическими аспектами, помогающими использовать, адаптировать и моделировать их.

Предлагаемые ядро понятий и язык маленькие и легкие в своей основе, но они могут быть расширены, чтобы использоваться в более сложных ситуациях, касающихся систем, критических в отношении жизни, надежности, бизнеса, своих задач и безопасности.

### 3.4. Реализация ядра

Эти требования амбициозны, но у нас достаточно свидетельств того, что цель может быть достигнута. Множество видных

приверженцев инициативы Semat при присоединении говорили нам, что у них есть искомое ядро понятий, или что они видели примеры таких ядер, создаваемых в разных компаниях. Более того, известно, что можно разработать ядро понятий для определения практик.<sup>2</sup> Эта работа служит иллюстрацией того, что наши требования разумны и могут быть реализованы.

Хотя каждое из этих существующих ядер может предоставить часть ответа, который ищет Semat, ни одно из них не общепринято, что является очень важным для полного успеха нашей инициативы.

Разработка ядра – это не только техническая проблема. Прежде всего, это вопрос достижения соглашения о том, каковы должны быть базовые элементы ядра и что они должны в себя включать, например, должны ли таковыми считаться состояния и условия переходов между состояниями.

### 1. Понятие ядра

Ядро не должно быть ни новой унифицированной методологией, ни новой метамоделью процессов, ни новой совокупностью знаний, ни новым языком моделирования, ни трюком, заставляющим создавать или покупать больше инструментов. Ядро должно быть просто, как набор понятий для того, что у нас уже есть (например, команды и проекты), что мы уже делаем (например, специфицируем и реализуем), что мы можем получить (например, программные системы), когда мы разрабатываем ПО, независимо от того, как мы работаем, пишем ли мы документацию или нет, даже от того, является ли результат хорошим или плохим. Ядро должно быть точным, концентрированным и легким. Более подробное объяснение см. в Приложении 3.

### 2. Управление разработкой ядра

Недавно, для того чтобы обеспечить необходимое управление разработкой ядра, эта работа была передана в Группу управления объектами (Object Management Group, OMG, <http://www.omg.org/>). Это перемещение гарантирует открытость и справедливость процесса выбора и то, что

<sup>2</sup>см. [ftp://ftp.ivarjacobson.com/outgoing/kernel/Software\\_Development\\_Kernel.pdf](ftp://ftp.ivarjacobson.com/outgoing/kernel/Software_Development_Kernel.pdf)



Рис. 1. Текущий набор предполагаемых элементов ядра. (Работа продолжается и возможны изменения.)

результаты принесут пользу всему сообществу. Запрос предложений (RFP), озаглавленный „Проблемно-ориентированный язык и ядро понятий для программной инженерии (ES-SENSE)“, был подготовлен группой членов OMG и представлен в OMG. RFP содержит призыв подготовить язык и ядро понятий, позволяющие быстро описывать элементы своих текущих и будущих практик и методов так, чтобы они могли быть скомпонованы, смоделированы, применены, сопоставлены, адаптированы, внедрены, оценены, измерены, исследованы и включены в обучение. Методы, практики и существенные элементы ядра должны описываться на данном языке.

RFP основан на концепции Semat, представленной год назад [4]. На следующем шаге OMG рассмотрит RFP, и после его одобрения в июне 2011, любая организация или команда сможет высказать свои предложения в ответ на RFP.

Группа из примерно 20 человек, работавших над проектом ядра понятий в Semat, начиная с марта 2010, будет одним из участников этой деятельности. Эта группа объединяет людей со всех концов света, представляющих практиков, руководителей, преподавателей и исследователей.

### 3. Текущий статус разработки ядра

Рабочая группа Semat на сегодняшний день имеет первую версию ядра понятий. Она будет проверена небольшой группой потенциальных пользователей, отзыв которой покажет, находится ли работа на правильном пути. На сегодняшний день ядро включает набор исходный версий понятий, включающих „вещи, которые мы производим и развиваем“, „деятельности, которыми мы занимаемся“, и „навыки, которые мы должны иметь“. Например, следующие понятия рассматриваются как существенные элементы этого ядра: бизнес-возможность, требования, программная система, команда, работа, метод выполнения работы и практика (рисунок 1). Эти элементы имеют четко определенные состояния.

Например, требования могут быть осознаны, в процессе обсуждения, устойчивы, корректны, подготовлены для проверки и выполнены. Продолжается работа над „деятельностями, которыми мы занимаемся“, и „навыками, которые мы должны иметь“. Это ядро в конечном итоге поможет практикам при оценке развития и общего состояния их проектов в сравнении с желательными целями. Практики будут применять его, чтобы оценить свои

текущие практики и чтобы расширить их для применения в особых условиях.

Однако, ожидается, что будут предложены и другие варианты ядра понятий. При наличии нескольких предложений стандартная процедура OMG состоит в организации совместной работы различных участников над общим предложением, которое в итоге будет принято. Ожидается, что окончательно принятое OMG ядро будет стабильным, но не будет застывшим. Оно продолжит развиваться по мере того, как наше понимание программной инженерии будет углубляться, а область расти.

#### 4. Продолжение текущей работы

Так как на сегодняшний день у нас нет общепринятого ядра (того, которое, как ожидается, будет принято OMG), и так как существование ядра является основой Semat, как же тогда сегодня мы можем создать план на три года вперед для Semat?

Продолжение работ в Semat, конечно, будет зависеть от результатов работы OMG, но значительная их часть должна выполняться из-за одного того только, что мы движемся вперед, к общепринятому ядру понятий.

Например, следующие работы, некоторые из которых уже начаты, в действительности не зависят от деталей ядра:

- формулирование требований к Semat на основе различных потребностей пользователей (работа уже началась);
- оценка того, что работа OMG отвечает требованиям групп пользователей;
- обработка отзывов от групп пользователей;
- поддержка доступности результатов;
- разработка практических теорий, покрывающих общественные и технические потребности, соответствующих интересам бизнеса, исследователей и практиков, и проверяемых как эмпирически, так и формально;
- сертификация практик. (Semat не будет одобрять и отклонять различные концепции практик, но будет гарантировать повторную используемость определений практик).

Для успеха любого принятого OMG ядра также очень важно, чтобы работа продолжалась и вне стандартной площадки, которую обеспечивает OMG; это должно гарантировать, что результаты OMG соответствуют запросам пользователей.

#### 5. Работа после принятия ядра OMG

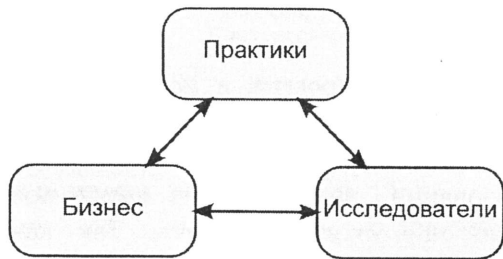
После того, как OMG примет некоторый вариант ядра, будет еще много работы, связанной с тем, что ядро будет стабильным, но не застывшим. Поэтому, необходимо параллельно работать над реализацией ядра и над его применениями в реальных проектах.

Есть другая, зависящая от конкретики ядра, работа, например, спецификация переиспользуемых практик. Предполагается, что практики будут описаны как в виде текста, что, конечно, может быть сделано уже сегодня, так и с помощью разрабатываемого языка. Таким образом, часть работ может начаться только тогда, когда язык будет определен. Когда мы узнаем больше о деталях ядра, нам нужно будет обновить часть результатов, но мы предполагаем, что в конце трехлетнего периода мы установим набор и вид продуктов Semat, что обсуждается далее в Разделе 3.1.

#### 3.5. Использование ядра с меняющимся уровнем детализации

Ядро легковесно, поэтому его легко понять и использовать как словарь или как карту для определения практик и целых методов. Практики могут быть определены на любом уровне детализации, есть два крайних варианта: очень неформальный и очень строгий. Во многих случаях будут использоваться промежуточные варианты.

Для маленьких команд иногда совершенно правильно сохранить практики как неявное знание, которое развивается посредством общения в команде. Для команд, работающих с системами, критическими в отношении бизнеса, выполняемых задач, надежности и безопасности, требуется большая строгость, и практики, используемые ими, должны быть определены на языке с формальной основой. Таким образом, язык ядра должен поддерживать несколько стилей определения практик.



**Рис. 2.** Сообщество, связанное программным обеспечением, и его самые важные группы пользователей.

#### 4. ПОСЛЕ ПЕРВОГО ЭТАПА – ДВИЖЕНИЕ ВПЕРЕД С SEMAT

Общепринятое ядро понятий и язык являются важнейшими базовыми компонентами и необходимыми инструментами, чтобы переосмыслить программную инженерию и чтобы утвердить общее основание. „Сейчас еще не конец. Это даже не начало конца. Но, возможно, это конец начала“.<sup>3</sup> Это первый шаг, который позволит нам расширить работу, даже до того, как реализации ядра и языка станут общедоступными.

Главная наша цель – сделать сообщество, связанное с разработкой программного обеспечения, успешным в целом. Это сообщество огромно, и для того, чтобы достичь нашей цели, мы должны сконцентрироваться на самых важных группах пользователей, а именно, на практиках, бизнесе и исследователях (рисунок 2). Нам нужно добиться их успеха, как по отдельности, так и совместного. Простыми, но не исчерпывающими словами: практики и бизнес определяют потребности, которые нужно изучать и исследовать. Ученые исследуют и формализуют те потребности, которые должны быть лучше поняты. Практики – это деятели в рамках бизнеса, а бизнес направляет практиков при создании бизнес-ценностей.

Это широкие группы пользователей с множеством разнообразных интересов в области программной инженерии. Тем не менее, Semat был и всегда будет открытой инициативой, учитывающей интересы разных групп. Упрощение структуры групп пользователей

необходимо для обсуждения и выработки ключевых целей и показателей.

В течение следующих трех лет участники Semat будут разрабатывать набор продуктов, нацеленных на достижение успеха.

##### 4.1. Продукты

Участники Semat создадут целый спектр продуктов, чтобы поддержать свое видение программной инженерии. Предполагается, что эти продукты будут выпущены по отдельности в различные моменты времени в течение следующих трех лет.

*Ядро и язык.* В рамках OMG будут созданы ядро и язык, основывающиеся на общем основании программной инженерии. Их публикация сделает их общеизвестными и подготовит возникновение новой экосистемы для методов и практик, основывающихся на открытых стандартах. Эта работа не должна быть. Ядро понятий будет включать общепринятые важные вещи, а также те, что используется практиками в каждом проекте при создании ПО. Ядро будет содержать терминологию, способствующую пониманию и взаимодействию по поводу общепринятых вещей, касающихся требований, команд, программных систем, заинтересованных лиц, работ, бизнес-возможностей и измеримых показателей.

*Инструменты.* Набор инструментов (включая инструменты с открытым исходным кодом) – или в качестве отдельных инструментов, или как встраиваемые модули для существующих инструментов – станет доступен, что позволит людям разрабатывать, обозревать, компоновать, сравнивать, подвергать сомнению, измерять и использовать различные практики и методы. Инструменты для различных проблемных областей программной инженерии смогут взаимодействовать на основе общих понятий и языка. Эти инструменты будут нацелены на реализацию нужд практиков, чей основной интерес состоит в создании качественного ПО, удовлетворяющего клиентов.

*Рынок практик.* Открытые стандартные язык и ядро сделают возможными публикацию, классификацию и обмен практиками. Рынок предоставит среду, где разработчикам будет

<sup>3</sup>Сэр Уинстон Черчилль, выступление 10 ноября 1942 года.

дана свобода использовать предпочитаемые ими техники в различных контекстах. Он станет тем местом, где проверенные практики и новые инициативные идеи будут легко доступны.

*Учебные программы.* Появится новая и более систематическая основа для образования в области программной инженерии, что будет способствовать повышению квалификации в академической и профессиональных средах. Будут созданы основанные на ядре, языке, практиках и методах учебные программы по информатике и программной инженерии для использования в университетах, а также в курсах для исследователей, инженеров-электриков, инженеров в машиностроении и других, чьи профессиональные знания находятся в разнообразных смежных дисциплинах.

*Учебники и статьи.* Новые учебники и справочный материал для учебных программ и личного развития, основанные на ядре и языке, будут разработаны и сделаны общедоступными. Будет написано много книг по практикам, определяемым с помощью ядра, которые будут предназначены для пользователей разного уровня, чтобы поддержать практиков при улучшении их навыков.

*Исследования.* Объективность, сравнимость практик, а также возможность внедрить, использовать, адаптировать и моделировать их, приведут к возрождению исследований в программной инженерии. У исследователей появится общая инфраструктура, которая будет служить в качестве испытательного стенда и поможет быстро реализовывать новые идеи (расширяемые практики).

#### *4.2. Пояснительный пример: команда в маленькой компании, разрабатывающей одно приложение*

Следующий сценарий иллюстрирует варианты использования продуктов Semat.

Представьте руководителя проекта, собирающегося начать новый проект, чтобы разработать новую версию существующего продукта.

У его команды есть некоторый метод работы, но он не обязательно документирован. Все участники знают о ядре Semat. Они хотят изменить свою работу, чтобы приобрести большую гибкость, в частности, они хотят

улучшить методы работы с требованиями и тестами.

Руководитель проекта и команда начинают работать на основе ядра. Они делают эскиз (например, нечто похожее на модель вариантов использования) некоторых из своих практик, которые они хотят сохранить. Так как они знают свой способ работы, это делается быстро. В основном, они просто работают со своими старыми терминами и синхронизируют их с элементами ядра.

Затем они идут в библиотеку практик своей компании и выбирают такие практики, которые наилучшим образом отвечают их потребностям. Они скачивают инструмент, помогающий им понять новые практики (возможно их существующие практики также будут кратко схематично изображены). Если нужно, они адаптируют практики, чтобы те подходили для их специфичных потребностей (в этом сценарии адаптация очень ограничена). Инструмент также помогает им использовать практики в проекте, итерации и после итерации. Им также нужны (другие) инструменты, чтобы поддерживать свои новые практики, но и старые инструменты все еще будут работать в рамках существующих практик.

Положительный побочный эффект практик состоит в том, что построенные на их основе учебные материалы очень эффективны. Библиотека практик может пополняться с рынка практик Semat, поэтому ее учебные материалы создаются и используются во всем мире, а также много раз улучшаются. Некоторые виды обучения могут осуществляться удаленно.

Команда поддерживает изменения, так как она легче воспринимает общее состояние проекта; в любой момент она знает, в каком положении проект и куда он движется; точно известно, когда и что было сделано. Команда также довольна почти полным отсутствием сверхурочной работы.

Руководитель проекта удовлетворен заметно лучшими результатами, меньшим и более предсказуемым временем выхода на рынок, более низкими затратами, чем предполагалось, а также тем, что клиенты довольны. И теперь другие команды хотят переиспользовать этот опыт, поэтому руководитель также успешен.





Рис. 3. Цели ЛБУ и связанные характеристики.

Для иллюстрации некоторые сценарии для основных групп пользователей – практиков, бизнеса и исследователей – представлены в Приложении 4.

#### 4.3. Показатели успеха для пользователей

Успех Semat будет измеряться по его способности позитивно влиять на развитие программной инженерии с точки зрения практиков, бизнеса и исследователей. Лучше, быстрее и успешнее (ЛБУ; Better, Faster and Harrier) – это метрики, обращенные как к общим целям, так и специфическим интересам этих отдельных групп.

В широком смысле, „лучше“ подразумевает меньше возможностей для совершения ошибок, меньше переделок и более высокую эффективность удаления ошибок, чем сегодня; „быстрее“ подразумевает более короткие циклы разработки, чем сегодня; в то время, как „успешнее“ означает повышение удовлетворенности клиентов и работников на протяжении многолетнего жизненного цикла ПО.

В особых случаях ЛБУ имеет различные значения для трех групп пользователей, на которых мы сосредоточились.

1. Для практика „лучше“ означает, что его

конкурентоспособность выше, он разрабатывает лучшее программное обеспечение и имеет опыт использования большего числа практик. „Быстрее“ значит, что он учится и делает свою работу быстрее. „Успешнее“ значит, что он больше уверен в своих силах и может легко переходить из одной организации в другую.

2. Для бизнеса „лучше“ значит, что он инновационен, выпускает программное обеспечение высокого качества и использует объективные показатели состояния проекта. „Быстрее“ означает сокращение сроков поставок. „Успешнее“ подразумевает удовлетворенных клиентов и сотрудников.

3. Для преподавателей „лучше“ образование означает знающих студентов, использующих надежные понятия и теории, которые отвечают требованиям бизнеса, а также использование целостной программы обучения, независимой от увлечений, но открытой для инноваций. „Быстрее“ значит более быстрое обучение и переход студентов в бизнес. „Успешнее“ означает более целеустремленных студентов и более привлекательное обучение.

Для ученых „лучшие“ исследования под-

разумевают концентрацию на областях, более тесно связанных с задачами практики. „Быстрее“ означает более быструю передачу технологий в бизнес и более быструю обратную связь от него. „Успешнее“ означает, что исследования влияют на качество программных продуктов.

#### 4.4. Измерение успеха

##### 1. Качественные показатели

Нам нужно определить, как можно четко связать продукты Semat с целями практиков, бизнеса и исследователей. Эти цели могут быть качественно измерены с помощью ЛБУ – более точно, за счет характеристик ЛБУ.

ЛБУ – это цели, которые мы хотим достичь в рамках сообщества разработчиков ПО. У них есть более конкретные и измеримые характеристики (см. рисунок 3). По мере того, как Semat движется вперед, мы будем выделять и обсуждать такие характеристики. Мы – применяя принцип разделения ответственности – отделим общие для всех целевых групп характеристики от тех, которые предназначены отражать специфические потребности различных пользователей.

Некоторые средства оценки качественных показателей включают определение реакции пользователей, например, с помощью интервью и вопросников из комментариев, мнений, оценок и взглядов на продукты. Мы, скажем, можем собрать мнения каждой группы пользователей (практиков, бизнеса и исследователей) о том, как наши продукты влияют на заданные характеристики. Как пример, мы можем оценить, насколько выросла конкурентоспособность практика, использующего наши продукты.

Чтобы продемонстрировать наш подход ЛБУ, мы выделили ряд характеристик и привязали их к целям ЛБУ на рисунке 3. Этот пример, однако, не следует воспринимать, как полноценную и готовую к использованию схему. Мы предполагаем, что из этих свойств наиболее существенными являются: объективные показатели состояния, конкурентоспособность и удовлетворение.

Поэтому они будут конкретизированы в Приложении 5.

##### 2. Количественные показатели

Можно измерить влияние продуктов Semat количественно, измеряя их прямое влияние на создаваемое ПО. Например, измеряя сокращение цикла разработки, ROI (коэффициент окупаемости инвестиций), уменьшение количества ошибок и переделок, повышение надежности и удовлетворенности клиентов. Эти количественные показатели обеспечивают объективную оценку успеха. Оценка итогов в далекой перспективе требует эмпирических исследований в организациях, внедривших результаты нашего проекта, что будет одной из его долгосрочных задач.

Другой способ количественной оценки результатов заключается в измерении использования наших продуктов сообществом, связанным с разработкой программного обеспечения. Является ли эта метрика адекватной или нет остается предметом обсуждения при развитии проекта, но это, конечно, интересный и важный показатель, потому что широкое принятие его продуктов является очень важным для успеха. Фактическое количество внедрений должно браться не из воздуха, а из успехов практиков, бизнеса и исследователей в использовании продуктов.

Мы планируем, что в течении трехлетнего периода (но открыты и для других предложений по показателям):

1. Более 30000 практиков будут использовать в своей ежедневной работе продукты, созданные под влиянием Semat;
2. Более 50% всех университетов с кафедрами информатики будут использовать ядро и язык в учебной программе по программной инженерии, студенческих дипломах и диссертациях;
3. 80% компаний из списка Fortune 500 будут иметь по меньшей мере один активный проект, применяющий понятийное ядро и язык;
4. В проектах с открытым исходным кодом на основе Semat будет более 100 активных участников;

5. Один из основных поставщиков ПО (или более) поддержит ядро и язык Semat;
6. Две основных аутсорсинговых компании (или более) будут использовать ядро и язык Semat, чтобы описывать свои практики.

Эти цифры основываются на оценке того, что нужно для объявления успеха. Хотя они не основаны на каком-либо эмпирическом материале, они также не совсем случайны. Эти показатели достижимы, так как у двух ведущих поставщиков решений по процессам разработки ПО значительно больше пользователей, чем указано в этих оценках. Главные отличия состоят в том, что их решения не имеют общего основания (или ядра) и используют специфичные языки – оба являются очень важными целями для Semat. Ясно, что получение поддержки ведущих поставщиков важно, но она возможна только вследствие поддержки их основных клиентов. Мы получили поддержку от нескольких крупнейших корпораций, но для движения вперед нам все еще нужно делать гораздо больше.

## 5. ЗАКЛЮЧЕНИЕ

В настоящее время Semat находится на пути реализации Великой мечты, которую ведет все сообщество. Предстоящий путь будет осложнен проблемами расширения его границ. Проблемы – это также и возможности. В случае успеха Semat значительно изменит сообщество, связанное с ПО, и даст ему новую площадку, чтобы создавать более хорошее, более быстрое и более успешное программное обеспечение. Уотс Хамфри продемонстрировал, возможно, настоящую предусмотрительность перед первым собранием Semat в Цюрихе в марте 2010, когда сказал: „Вероятно, собрание в Цюрихе будет историческим событием, как сессия НАТО 1968 года в Гармише“. Результаты Semat – это коллективная работа сообщества на благо всего сообщества. Сотрудничая, мы можем переосмыслить программную инженерию.

### Благодарности

Авторы хотели бы поблагодарить следующих людей, которые внесли вклад в создание данной статьи и дали ценные отзывы: Дэйв Канингем,

Брайан Элвсетер, Майкл Годик, Уинфред Менезеш и Иэн Спенс.

Ниже приведены пять приложений, поясняющих некоторые темы, затронутых в статье.

### Приложения

#### Приложение 1: Краткая история Semat.

В конце 2009 г. Айвар Джэкобсон, Бертран Мейер и Ричард Соли (неформально известные как „тройка“) выдвинули инициативу по переосмыслению программной инженерии как строгой науки. Они признали, что естественное развитие в этой области идет путем наименьших изменений для приближения к корректности создаваемого ПО, но этим путем нельзя идти, если мы хотим поддержать ИТ-индустрию и помочь ей соответствовать интересам общества. „Тройка“ провозгласила необходимость переосмысления программной инженерии на прочной основе, беря лучшее из всех достижений в теории и практике за последние пятьдесят лет. В призыве к действиям они описали стоящие проблемы и очертили путь, по которому следует двигаться вперед, – Великую Мечту.

#### Призыв к действиям

Развитие программной инженерии сегодня серьезно затруднено незрелыми практиками. Специфические проблемы этого развития включают следующие:

- распространенность увлечений, более свойственное индустрии моды, чем инженерной дисциплине;
- отсутствие прочных общепринятых оснований;
- огромное число методов и их вариаций, разница между которыми плохо понимается и искусственно преувеличивается;
- отсутствие заслуживающих доверия экспериментальных оценок и подтверждений;
- разделенность промышленной практики и академических исследований.

Мы поддерживаем процесс переосмысления программной инженерии, основанный на строгой теории, проверенных принципах и наилучших практиках, который:

- включает ядро общепринятых понятий, расширяемое при использовании в специфических ситуациях;
- учитывает как технологические вопросы, так и человеческий фактор;
- поддержан бизнесом, исследователями и пользователями;
- расширяем для поддержки изменяющихся требований и технологий.

Тройка была обрадована и польщена, узнав, что по прошествии небольшого времени десятков академических и промышленных организаций и около трех дюжин известных в области программной инженерии и информатики людей поддержали высказанные тезисы. Более того, они были поддержаны еще более чем 1400 специалистами.

Концепция Semat (<http://www.Semat.org/pub/Main/WebHome/SEMAT>) описывает понимание проблем программной инженерии тройкой, возможности для их решения и план первого этапа проекта. С момента публикации в феврале 2010 г., более 20 исследователей, хорошо знакомых и с ИТ-бизнесом, добровольно внесли значительный вклад в претворение этого плана в жизнь, т.е. в реализацию общей идеи методов разработки, основанных на строго описанных, четко понимаемых, сопоставимых практиках, определяемых на основе ядра базовых понятий на специальном проблемно-ориентированном языке.

#### **Приложение 2: Разделение ответственности**

Semat применяет принцип разделения ответственности.<sup>4</sup>

- Поддержка программных систем отделена от поддержки систем в целом (включающих аппаратуру) и решений (включающих аппаратуру и людей). Поэтому ядро и язык должны быть расширяемыми так, чтобы поддерживать системы и решения без усложнения их использования для ПО.
- Выделены, по крайней мере, два разных взгляда на процессы разработки: взгляд

инженеров-технологов и программистов-практиков. Основные пользователи методов и практик – это программисты-практики (разработчики, тестировщики, руководители проектов и пр.). Результаты Semat должны быть понятны и инженерам-технологам, и практикам, но при этом нацелены, в основном, на последних. Для эффективного использования результатов технологами предусмотрены специальные расширения, не затрудняющие работу практиков.

- Важные вещи отделены от несущественных; например, основное руководство от детализированного, или явное знание от неявного. Это позволяет инженерам-технологам создавать на такой основе легковесные масштабируемые методы. Иными словами, Semat сконцентрирован на самом важном – „наименьшем общем знаменателе“, который присутствует во всех успешных проектах в области ПО.
- Общие определения терминов отделены от специализированных деталей понятий, допуская не исключение, а включение результатов предшествующих методологических работ. Другими словами, мы ищем общую основу, на которой могут быть построены все существующие методы.

#### **Приложение 3: Ключевые понятия**

Ядро и его элементы будут определены на специальном проблемно-ориентированном языке (предназначенном для описания практик разработки ПО и методов).

Результат должен удовлетворять следующим требованиям:

1. Метод – это композиция практик (а не просто набор связанных компонентов процессов, методов, деятельностей и пр.).
2. Практика – это способ решения какой-то задачи с определенной целью. Существует несколько типов практик, но основные и самые важные – это конкретные практики. Конкретная практика – это деятельность с четко определенными началом и концом,

<sup>4</sup>Общую информацию о нем можно найти в [http://en.wikipedia.org/wiki/Separation\\_of\\_concerns](http://en.wikipedia.org/wiki/Separation_of_concerns).

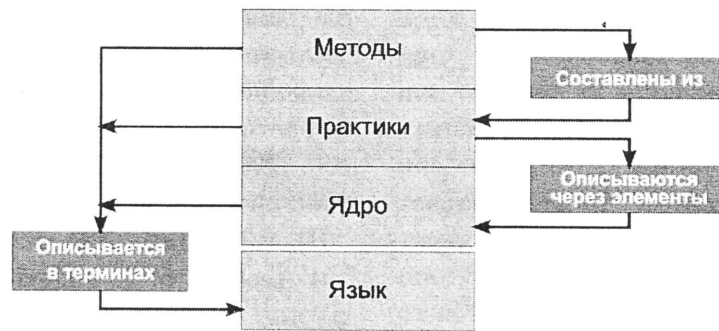


Рис. 4. Ключевые понятия.

которая используется программистами-практиками для решения их задач. Эти практики ценны по отдельности, на них хотят опираться пользователи, они должны быть измеримы с помощью специальных метрик – все это и есть необходимые характеристики практик как таковых.

3. Все методы имеют нечто общее – „общую основу“ или „суть программной инженерии“ – ядро базовых понятий. Примеры его существенных элементов: работа, команда, требования, программная система, бизнес-возможность, заинтересованные лица и пр.
4. Методам необходима теория – наша работа должна основываться на надежном теоретическом фундаменте. Методы komponуются из практик, а практики описываются в терминах их существенных элементов и в терминах других элементов, таких как деятельности и результаты. Формализация всех этих понятий в рамках некоторого языка – это отправная точка такой теории. Более того, многие практики могут быть формализованы или поддержаны формальными техниками – к примеру, использовать для измерений статистические модели.
5. Методы динамичны и используются часто. Методы – это не просто „чтение для разработчиков“, они должны быть изменяемыми, поддерживая ежедневную работу команды. Это меняет традиционное определение метода – теперь это не просто

описание того, что нужно делать, а определение реальных действий.

## Приложение 4: Некоторые сценарии использования ядра и языка

### П. 4.1. Сценарий для практика

Нынешние студенты, попадающие в коммерческие компании по окончании университета, хотя и могут применять некоторые приобретенные там навыки (например, C++ или Java), требуют существенного дообучения. Отчасти это неизбежно, к примеру, нужно изучать терминологию конкретной отрасли, но сегодня даже такие фундаментальные термины, как „требования“ или „команда“ по-разному используются в разных организациях.

Сегодняшнее общество очень мобильно, люди часто меняют работу на протяжении своей карьеры. Когда программист переходит в другую компанию – или даже в другое подразделение той же самой – он приносит с собой свой опыт, но все равно на новом месте многое нужно выучить заново. Это может настолько разочаровать человека, что он даже может думать о смене карьеры.

Установление ядра понятий не создает стандарт, отсекающий каких-то пользователей. Ядро, опирающееся на общее основание, поощряет создание новых подходов для конкретной работы, отталкиваясь от общепринятых вещей.

Что же это означает для практика? В будущем практики будут иметь больше доступных вакансий в силу своей большей мобильности без ущерба для удовлетворения от работы.

Они будут знать, что могут рассчитывать

на полученные в университете знания, когда будут переходить в промышленность, при переходе из одной компании в другую или при перемещении между проектами в рамках одной компании. Понимание того, что они изучили самое необходимое, также придаст им большую уверенность в себе и будет способствовать самореализации, поскольку они смогут потратить свое ограниченное время на то уникальное, что есть в их работе, а это выгодно их работодателю, клиентам, да и им самим.

#### П. 4.2. Сценарий для бизнеса

Ключевым знанием в любой организации является четкое понимание состояния проектов и знание, что надо предпринять в случае возникновения проблем. Эффективные объективные показатели помогают менеджерам понять, когда требуется вмешательство. Большое количество объективных данных свидетельствуют, что главные причины проблем в проектах связаны с неспособностью организаций обнаруживать проблемы на ранних этапах.

Компании часто внедряют или разрабатывают один метод для всех ситуаций, надеясь каждый раз находить решение в корпоративных стандартах. К несчастью, этот подход имеет малые шансы на успех, поскольку не учитывает критически важные факторы, различные для разных проектов, команд, продуктов, организаций и пр. Но есть ли лучший подход?

Менеджеру проекта, работающему в промышленной разработке ПО, ядро предоставит средства для определения целостного набора важнейших показателей для всех его проектов, вне зависимости от их размера или вида. Распределенная разработка довольно распространена во многих больших компаниях, и ядро поможет, в числе прочего, сделать общение между географически разделенными разработчиками более эффективным.

Лидеру команды, который определяет способ работы над новым проектом, ядро поможет выбрать подходящий для команды путь достижения целей организации. Другим преимуществом будет простота распределения людей по проектам (даже в разных областях), поскольку у них будет общий базис, на котором они строят свою работу.

Технологам ядро поможет понятно объяснять,

как работает компания, новым и потенциальным работникам. Это упростит использование сторонних наработок, переиспользование и предоставит командам способы интегрировать их общие знания в должным образом адаптированные процессы разработки.

#### П. 4.3: Сценарий для исследователя

В рамках сегодняшней системы образования разные университеты и профессора имеют разные требования и интерпретации, касающиеся преподавания программной инженерии и содержания соответствующих программ. Есть некоторые рекомендации для сертификации. Например, Комитет по сертификации инженерии и технологии (Accreditation Board for Engineering and Technology, ABET) в США и Рекомендации по составлению университетских программ в программной инженерии (SWEBOK Curriculum Guidelines for Undergraduate Degree Program in Software Engineering). К сожалению, эти рекомендации обычно очень абстрактны и оставляют детали программ на усмотрение университетов и профессоров. Отсутствие базового ядра понятий программной инженерии приводит к наличию большого числа подходов к обучению, которые не имеют ясной теоретической базы.

Это приводит к тому, что студенты, получившие образование в разных местах, имеют разные наборы навыков, что не соответствует нуждам бизнеса. Это также приводит к погоне за модой в исследованиях, а не следованию четкому сбалансированному плану.

Для преподавателей ядро, построенное на общем основании, означает базу для а) преподавания программной инженерии; б) разработки курсов программной инженерии; и в) демонстрации студентам плюсов и минусов разных методов работы. Такое ядро гарантирует единообразное представление важнейших составляющих программной инженерии в различных университетах и образовательных программах.

С точки зрения исследователя ядро предоставляет единые, связанные с реальной практикой, способы проведения экспериментов в рамках различных подходов к программной инженерии, и крепкую основу для отделения пустой рекламы от работающих методов.

## Приложение 5: Примеры характеристик ЛБУ

Как показано на рисунке 3, изначально мы предложили десять характеристик для цели „Лучше“, девять для „Быстрее“ и шесть для „Успешнее“. Двигаясь дальше, мы, скорее всего, увидим, что ряд характеристик будут влиять сразу на несколько целей ЛБУ, но на сегодняшний момент они являются достаточно иллюстративным приближением к нужному набору показателей.

Ниже рассматриваются три выделенных характеристики, и их возможное значение для наших групп пользователей: практиков, бизнеса и исследователей.

### • **Объективные показатели состояния проекта**

**Для практика:** дают средства формулировать объективные оценки работы, ее результатов и персонального развития. Общее основание дает возможность принимать грамотные решения о построении и переиспользовании практик. Можно критически осмысливать инновационные идеи и, тем самым, объективно оценивать инновации в индустрии, чтобы понять, пойдут ли они на пользу или во вред. Опыт, базирующийся на надежной и осознанной практике, даст новые, инновационные идеи для улучшения способов работы.

**Для бизнеса:** дают возможность лучше управлять компанией. Общее основание позволит вам поддерживать нацеленность команд на бизнес-возможности через выбор подходящих методов работы, сохраняя при этом корпоративный контроль. Можно критически осмысливать инновационные идеи и, тем самым, объективно оценивать инновации в индустрии, чтобы понять, пойдут ли они на пользу или во вред. Опыт, базирующийся на надежной и осознанной практике, даст новые, инновационные идеи для улучшения способов работы.

**Для исследователя, преподавателя, студента:** дают больший контроль над исследованиями, педагогическим подходом

или обучением, соответственно. Общее основание позволит вам принимать обоснованные решения о направлении вашей работы. Можно критически осмысливать инновационные идеи и, тем самым, объективно оценивать инновации в индустрии, чтобы понять, пойдут ли они на пользу или во вред. Опыт, базирующийся на надежной и осознанной практике, даст новые, инновационные идеи для улучшения результатов.

### • **Конкурентоспособность**

**Для практика:** конкурентоспособность повышается, поскольку работа выполняется быстрее и получаются более качественные результаты. Следовательно, становится проще вырабатывать общие навыки и достижения с коллегами, и тем самым лучше работать вместе и развивать команду.

**Для бизнеса:** конкурентоспособность повышается, поскольку работа выполняется быстрее, уменьшается длительность производственного цикла, и тем самым создается возможность занять лидирующие позиции на рынке.

**Для исследователя:** поскольку исследования строятся на общем основании, получаемые результаты надежны и обоснованы; повышается конкурентоспособность и привлекательность для текущих и потенциальных спонсоров.

**Для преподавателя:** поскольку преподавание основано на общем основании, повышается качество образования, конкурентоспособность и привлекательность для студентов.

**Для студента:** поскольку знания основаны на общем основании, создается стабильный фундамент для развития, исследований нового и непредвзятого, убедительного распространения новых идей. Повышается конкурентоспособность и привлекательность для будущего или текущего работодателя.

- **Удовлетворение.** Для практика: появляется уверенность в наличии

необходимых для продуктивной работы знаний и в соответствующей внешней оценке на рынке труда. Повышается благополучие, поскольку компетентность находит понимание и хороший прием в различных организациях, что также дает большую мобильность.

**Для бизнеса:** повышается уверенность в том, что продукты и компетенции персонала развиваются целостно и с лучшими результатами. Интересы организации приводятся в согласие с интересами клиентов, партнеров и сотрудников, таким образом, клиенты и сотрудники становятся более довольны.

**Для студента:** повышается уверенность в том, что получаемые знания построены на стабильной, непротиворечивой и общепринятой основе, доступно получение работы в любой компании мира.

## СПИСОК ЛИТЕРАТУРЫ

1. *Jacobson I., Meyer B.* Methods need theory. Dr. Dobb's J., August 06, 2009. <http://www.drdoobbs.com/architecture-and-design/219100242>
2. *Jacobson I., Spence I.* Why we need a theory for software engineering? Dr. Dobb's J., October 02, 2009. <http://www.drdoobbs.com/architecture-and-design/220300840>
3. *Jacobson I., Meyer B., Soley R.* Call for Action: The Semat Initiative. Dr. Dobb's J. December 10, 2009. <http://www.drdoobbs.com/architecture-and-design/222001342>
4. *Jacobson I., Meyer B., Soley R.* The Semat Vision Statement. <http://www.semat.org/pub/Main/~/WebHome/SEMAT-vision.pdf>
5. *Huang S.* the 1st Semat Workshop report. [http://www.semat.org/pub/Main/~/SematZurichMarch2010/~/Zurich\\_meeting\\_report.pdf](http://www.semat.org/pub/Main/~/SematZurichMarch2010/~/Zurich_meeting_report.pdf)