

**От ремесла – к инженерии,
от мастерской – к заводу
программных систем**

И.Е. Ермаков,
*Директор по науке и образованию
ООО «Метасистемы»*

ООО «Метасистемы»

Технологический институт ОрёлГТУ

Проект OberonCore

Инженерная деятельность

- Предсказуемость результатов, т.е. свойств создаваемых продуктов.
- Обеспечивается применением систематических методов, научно обоснованных.
- Методы в той или иной степени опираются на математический аппарат.
- Методы подкреплены арсеналом инструментальных средств.

Особенности программной отрасли

- Нематериальность систем.
- Свобода инженера от физических ограничений.
- Структурная, иерархическая сложность, в общем, превосходит сложность обычной инженерии.
- Велика доля паразитной, избыточной сложности.
- Связь со спекулятивными течениями в экономике.

Природа ВТ и ИТ

- результат развития двух ветвей – счётной техники (с переходом на электронную элементную базу) и математики (появление теории вычислимости и теории кодирования);
- входное сырьё и выходной продукт – абстрактный объект (число);
- «замыкание» математических моделей обратно на физические устройства, «обратное моделирование»;
- имеются противоречия между теоретической и практической информатикой.

Программа как конструкция

Показатель зрелости программиста:

процент внимания, направленный не на текст программы, а на её свойства.

- логические утверждения о состояниях между операторами (предусловия, постусловия, инварианты);

- протоколы взаимодействия компонентов - порядок обмена вызовами, сообщениями.

«Грамматики на динамическое поведение».

Мировой опыт 60-х – 80-х гг.

- очень крупные и ответственные проекты (оборонка, авиация, транспорт, связь...).
- консенсус относительно требований к промышленным языкам программирования (стандарт – Паскаль-языки: Ada, Modula-2).
- формирование технологии программирования, понимания программной инженерии (книги – Брукс, Фокс, Sommerwill, Bass, Bar; Липаев, Терехов, Жоголев).

Кризис 90-х гг.

- взрыв массовости (персональная техника, малые задачи);
- резкий приток новых кадров, преимущественно – самоучек;
- ограничения настольного оборудования, не позволившие сразу применить «нормальные» инструменты (полноценные ЯВУ, например, Ada);
- результат – «одичание» массового программирования (конкретно – распространение Си-семейства языков).

Роль Оберона

- научно-инженерная школа тьюринговского лауреата Н. Вирта (Швейцария, ETH);
- Oberon - семейство языков и ОС (на базе ПК собственной разработки);
- одновременно и промышленные системы (развитие Модулы-2), и настольные (развитие концепции интегрированной операционной среды, начатой в Smalltalk);
- основа для осевых технологий 90-х (Java, .NET, Google Go).

Выход

Для преодоления «болезней роста» в ИТ нужно создание качественного и устойчивого контура **наука – образование – производство – наука.**

Важнейшая роль в контуре – программная инженерия (решение сложных и ответственных задач, связь с «реальным сектором»).

Чистые исследования – прерогатива естественных наук. В области информатики, как и математики, без связи с реальными задачами, происходит вырождение.

Последние инициативы

- IEEE SWEBOK – «Свод знаний по программной инженерии» (*есть сайт с русским переводом*).
- SEMAT – «Software Engineering Method and Theory» - сообщество и Интернет-площадка (semat.org).
- Особое место – проект «Информатика-21».

Информатика-21

<http://www.inr.ac.ru/~info21>

- инициирована физиками (ИЯИ РАН) и инженерами (авионика, ПО для космических аппаратов);
- изучение ситуации в ИТ как наиболее яркого проявления проблем умственной деятельности (вопрос «как же может происходить такой маразм»);
- формулирование Принципа Калашникова – «Избыточная сложность – всегда уязвимость», и теории двух уровней интеллекта (комбинаторный, «приматический», усложняющий И абстрагирующий, рефлексирующий, упрощающий).

Информатика-21

<http://www.inr.ac.ru/~info21>

- задача единой, сквозной, фундаментальной системы преподавания программирования (5-й класс – 3-й курс ВУЗов, далее – наложение профильных спецкурсов);
- понимание долгосрочности образовательных требований (нужно закладываться на 10-20 лет);
- требованиям идеально удовлетворяют только Оберон-системы (конкретно - Компонентный Паскаль / среда BlackBox);
- огромная важность обучения и инструмента программирования для прикладников (инженеры, физики, математики, лингвисты, медики..);
- в 2009 г. выпущено 3 книги (партнёр – ДМК-Пресс).

Проект OberonCore

<http://OberonCore.ru>

- российский центр компетенции по Оберон-технологиям (г. Орёл);

- крупное форумное сообщество профессионалов из разных отраслей (ИТ, преподаватели, учителя, специалисты из аэрокосмической отрасли).

Завод программных систем?

- фабрика или лаборатория?
- Завод!

Завод программных систем:

- Применение систематических инженерных методов (в необходимой мере формальных).
- Предсказуемость свойств создаваемых систем.
- Обеспечение длительного цикла эволюции систем.
- Способность как решать сложные, наукоёмкие задачи (значимые для общества), так и выпускать «серийные» продукты.
- Наличие схемы передачи проектов от исследовательских и конструкторских групп к «производственным» (т.е. интеграция фабрики и лаборатории).
- Система устойчивого пополнения организации новыми кадрами.
- Способность выстраивать линейки продуктов, накапливать фонд базовых средств.

ООО «Метасистемы», наше видение

- Цель: не прибыльность, не продукт(ы), а существование и развитие коллектива, способного решать сложные задачи, выпускать технологии и линейки продуктов, накапливать фонд базовых методов и средств.
- С одной стороны - реальные задачи, с другой стороны – система обучения кадров.
- Необходимо развитие собственной инфраструктуры (на базе КП/ББ, своя middleware-платформа Oda Platform).
- Самый главный капитал профессионала – не продукты, не знания, а **правильное понимание природы явлений в своей области деятельности. Углубление понимания – превыше всего. => «Спешить некуда».**

Роль языка программирования

- с точки зрения долгосрочных приоритетов, выбор языка очень важен (устойчивое ядро для коллектива, в условиях разных задач и экспериментов с разными платформами, методами проектирования, и т.п.);
- принцип «выбрать один раз и забыть о проблемах»;
- использование разных языков, метание между ними препятствует накоплению в коллективе фонда базовых средств, арсенала инструментов, а это – важнейший капитал.

Оберон – Компонентный Паскаль

Причины большой эффективности:

- Сочетание **производительности** (компилируемости) и **динамичности** (сборка мусора, динамическая загрузка модулей, динамические средства типизации, рефлексия и метапрограммирования); тезис Ф.В. Ткачёва, из опыта применения в сложных задачах символьной алгебры.

Прекрасная замена популярной связке Си-Питон.

- Совершенно уникально сочетание **простоты**, **жёсткости**, **эффективности**; тезис И.Е. Ермакова, из опыта системного программирования.