

Анализ и проектирование на UML

Новиков Федор Александрович

fedornovikov@rambler.ru

Курс подготовлен по заказу
ООО Сан Майкросистемс СПб

Часть 8

Курс подготовлен при поддержке Sun Microsystems
Правила использования материалов опубликованы на www.sun.ru

План лекций

- Введение в UML
- Обзор языка
- Моделирование использования
- Моделирование структуры
- Моделирование поведения
- Управление моделями
- Тенденции развития языка
- ✓ UML и процесс разработки

8. Влияние UML на процесс разработки

- **8.1. Процесс разработки**
- **8.2. Повышение продуктивности**
- **8.3. Влияние UML**

8.1. Процесс разработки

- **Модель процесса**
 - = порядок проведения типового проекта
 - На уровне организации
- **Модель команды**
 - = отношения между людьми в процессе разработки
 - На уровне проекта
- **Дисциплина разработки**
 - = технология программирования (в широком смысле)
 - На уровне работника

Модель процесса

- **Фаза (phase)** — часть процесса работы над проектом. Обычно каждая фаза характеризуется вехой, достижение которой знаменует завершение фазы
- **Веха (milestone)** — одномоментное идентифицируемое событие, сопровождающееся появлением и фиксацией некоторого артефакта, который называется результатом вехи
- **Работник (worker)** — назначение сотруднику набора функций в рамках конкретного проекта
- **Артефакт (artifact)** — документ или иной материал, имеющий материальную форму и отчуждаемый от разработчика
- **Пример: Milestone oriented user-driven incremental software development process**

Стандартные процессы

- **Стандарты:**
 - **ISO 9000 (2000)**
 - **CMM (CMMI)**
 - **SPICE (ISO 15504)**
- **Модели:**
 - **RUP = Rational Unified Process**
 - **MSF = Microsoft Solutions Framework**
 - **PLCP = Product Life Cycle Process
от Sun Microsystems**
 - **...**

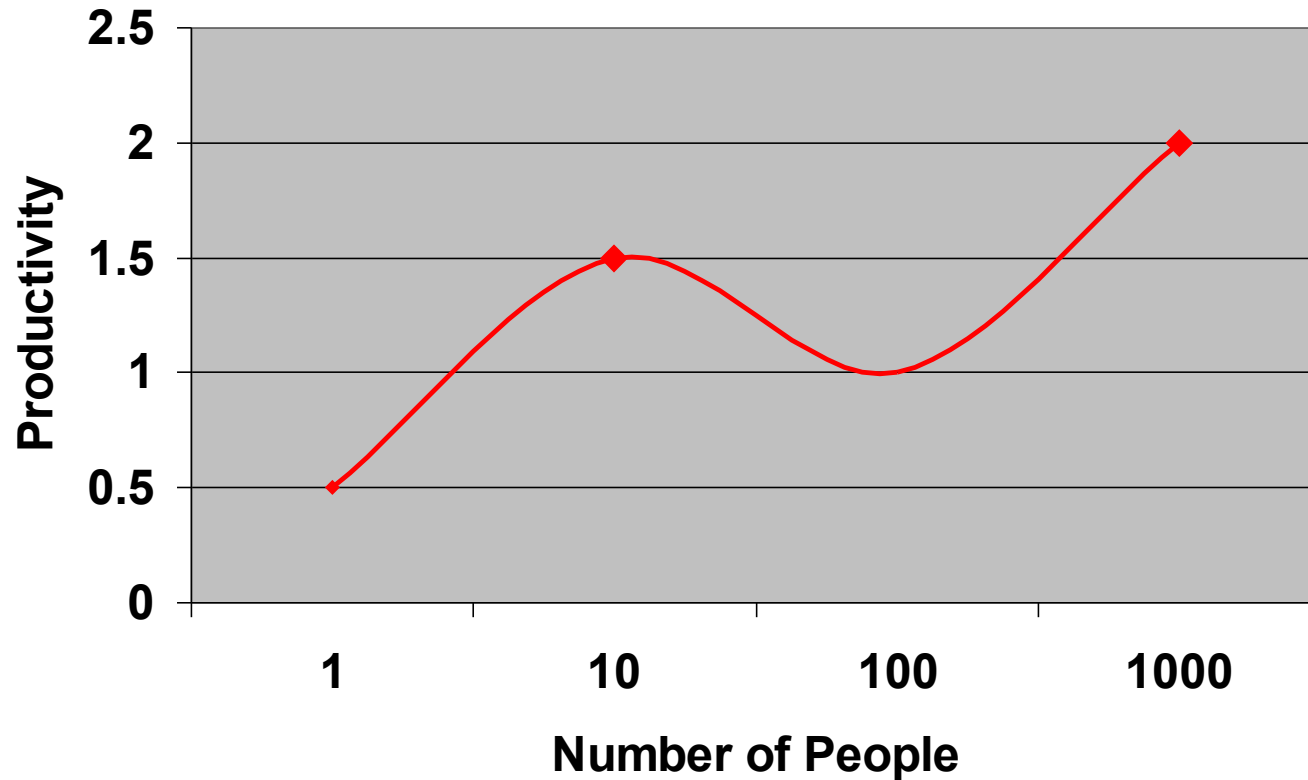
Типичный рецепт процесса

- **Взять дюжину свежих программистов среднего размера и хорошего качества**
- **Тщательно очистить, удаляя шелуху университетского образования**
- **Пассировать при средней температуре в течение месяца испытательного срока**
- **Поместить в толстостенный сосуд выбранной системы программирования**
- **Добавить менеджмента и теории по вкусу**
- **Варить в собственном соку при плотно закрытой крышке до готовности программного продукта, периодически выпуская релизы**
- **После употребления сохранять в темном помещении при комнатной температуре или использовать для приготовления следующего блюда**

8.2. Повышение продуктивности

- Программы ненадежны
- Программисты неуправляемы
- Программирование рискованно
- Продуктивность – производительность востребованного труда
- НЕ линейная зависимость продуктивности от различных факторов

Продуктивность и люди



Личная точка зрения автора. Диаграмма *качественная*.

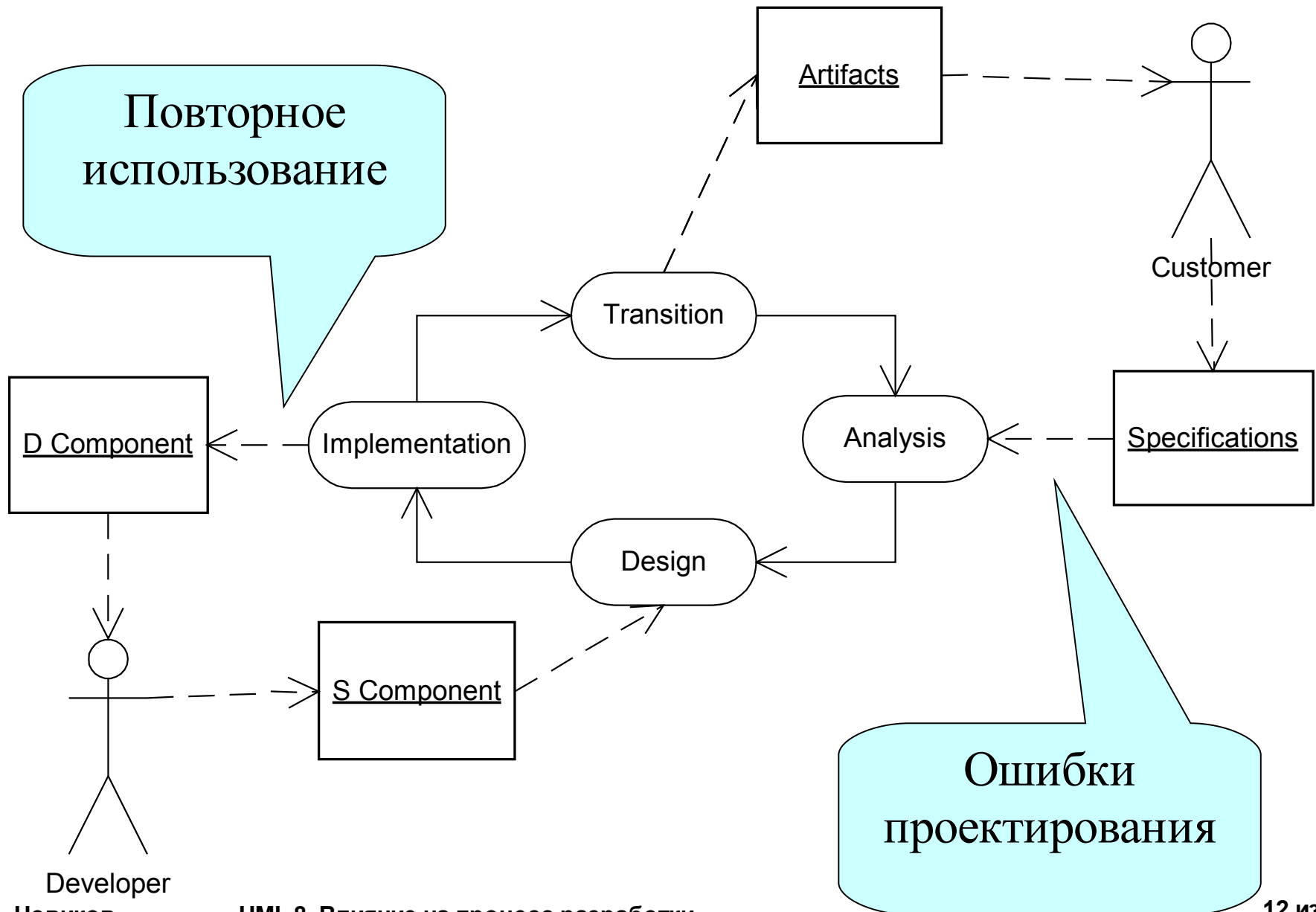
Повышение продуктивности

- **За счет УЛУЧШЕНИЯ процесса (x%)**
 - начальный процесс уменьшает продуктивность
 - best known practices
 - начиная с CMM 3 – управляемый процесс = измеримый процесс
- **За счет ОРГАНИЗАЦИИ команды (3x%)**
 - иерархия уменьшает продуктивность
 - динамическое переназначение ролей
 - управление по компетентности

Повышение продуктивности программирования

- За счет ТЕХНОЛОГИИ программирования (10x%)
- Сокращение количества внеплановых изменений кода
 - ошибки проектирования
- Увеличение объема повторно использованного кода
 - специфицированные компоненты в репозитории

Циклы повышения продуктивности



8.3. Влияние UML

- **UML → Специфицированные компоненты для повторного использования → скорость повторного использования возрастает → объем увеличивается**
- **UML → Согласованные спецификации для проектирования и реализации → число ошибок проектирования уменьшается**
- **Во всех фазах и процессах используется один и тот же язык → затраты на обучение уменьшаются**

Рекомендации по внедрению UML

- Организация должна поставить **ИЗМЕРИМУЮ** цель внедрения UML
- Знать и применять UML должны **ВСЕ** участники процесса
- Должен использоваться корпоративный **РЕПОЗИТОРИЙ** образцов проектирования на UML
- Инструмент, поддерживающий UML, должен быть **ЛЁГКИМ** в использовании

Выводы

- **Стандарты очень полезны, но не универсальны – требуется подгонка**
- **Нет процесса, наилучшего для всех типов организаций и проектов, но для каждой организации и каждого проекта в отдельности – есть наилучший процесс!**
- **UML – унифицирует представления артефактов в циклах повышения продуктивности, тем и хорош**