

**Санкт-Петербургский государственный университет
информационных технологий, механики и оптики**



**Описание практической работы студентов (ЛП)
по дисциплине «Технологические подходы к
разработке программного обеспечения»**

Новиков Ф.А.,

к.ф.-м.н., доцент кафедры «Технологии программирования»

Санкт-Петербург

2007

Оглавление

Цель проведения семинаров	4
Порядок подготовки и проведения семинаров	5
Содержание семинаров по темам.....	7
Инженерия программных систем.....	8
Дискуссионные вопросы.....	8
Рекомендуемая позиция первого панелиста	8
Рекомендуемая позиция второго панелиста	9
Этапы жизненного цикла	10
Дискуссионные вопросы.....	10
Рекомендуемая позиция первого панелиста	10
Рекомендуемая позиция второго панелиста	11
Сравнение моделей процесса разработки	12
Дискуссионные вопросы.....	12
Рекомендуемая позиция первого панелиста	13
Рекомендуемая позиция второго панелиста	13
Документированные процедуры	14
Дискуссионные вопросы.....	14
Рекомендуемая позиция первого панелиста	14
Рекомендуемая позиция второго панелиста	14
Сравнение моделей команды разработки	15
Дискуссионные вопросы.....	15
Рекомендуемая позиция первого панелиста	15
Рекомендуемая позиция второго панелиста	15
Сравнение парадигм программирования	16
Дискуссионные вопросы.....	16
Рекомендуемая позиция первого панелиста	16
Рекомендуемая позиция второго панелиста	16
Сравнение программных архитектур	18
Дискуссионные вопросы.....	18

ЛП Системы представления знаний

Рекомендуемая позиция первого панелиста	18
Рекомендуемая позиция второго панелиста	19
Тенденции развития технологии программирования	20
Дискуссионные вопросы.....	20

Цель проведения семинаров

Лабораторный практикум по курсу «Технологические подходы к разработке ПО» выполняется в форме панельных дискуссий с обзорными докладами студентов нормативных документов и первоисточников.

Целью лабораторного практикума по дисциплине «Технологические подходы к разработке программного обеспечения» является:

- изучение студентами текстов нормативных документов (стандартов) и фундаментальных источников по технологии программирования;
- приобретение навыков критического анализа и сравнения различных технологий, моделей, методов и парадигм программирования;
- приобретение навыков коллективного обсуждения сложных методологических вопросов.

Порядок подготовки и проведения семинаров

Семинар проводится в форме панельной дискуссии. В дискуссии принимают участие преподаватель и студенческая группа.

При проведении панельной дискуссии на семинаре выделяются следующие роли.

- **Ведущий.** В обязанности ведущего входит соблюдение регламента, контроль времени и подведение итогов.
- **Панелист.** В обязанности панелиста входит предварительная подготовка к дискуссии по указанным преподавателем источникам. Панелист должен подготовить и провести краткое (10–15 минут), но содержательное выступление, в котором позитивным образом отразить, каким образом дается ответ на дискуссионные вопросы в рекомендованных источниках. Другими словами, панелист должен достаточно подробно и последовательно изложить одну из альтернативных точек зрения на обсуждаемый вопрос.
- **Участник.** Участники могут задавать панелистам вопросы по теме дискуссии и могут высказывать собственное мнение или оценку позиции панелистов. Вопрос не должен продолжаться более минуты, а выступление – пяти минут.

Распределение по ролям.

Обычно роль ведущего играет преподаватель. Допустимо назначить на эту роль студента с целью приобретения последним навыков ведения заседания и управления аудиторией.

Роль панелистов обычно играют студенты, но, в крайнем случае, роль панелиста может взять на себя преподаватель. Рекомендуется на каждую дискуссию назначать двух панелистов. Допустимо назначать одного панелиста (при недостатке времени на подготовку или недостатке дополнительных материалов) или трех панелистов (при наличии достаточного времени на подготовку и большого объема дополнительного материала).

ЛП Системы представления знаний

Участники выступают по желанию. В течение семестра каждый студент должен хотя бы один раз быть панелистом и не менее трех раз выступить в дискуссии.

Содержание семинаров по темам

Предлагаемый набор семинаров – панельных дискуссий – указан в следующей таблице.

№	Название семинара	Тема учебного курса
1	Инженерия программных систем	Тема 1. Технология программирования
2	Этапы жизненного цикла	Тема 2. Жизненный цикл программы
3	Сравнение моделей процесса разработки	Тема 3. Модели процесса разработки
4	Документированные процедуры	
5	Сравнение моделей команды разработчиков	Тема 4. Модели команды разработчиков
6	Сравнение парадигм программирования	Тема 5. Дисциплина программирования
7	Сравнение программных архитектур	
8	Тенденции развития технологии программирования	

Темы семинаров могут быть выбраны и другие, также можно менять рекомендованные позиции панелистов, если есть в этом необходимость, но количество семинаров следует оставить пропорциональным объему материала в лекционном курсе.

Инженерия программных систем

Совокупность известных технологических подходов к разработке программного обеспечения, или, несколько короче, технология программирования, является инженерной дисциплиной, входящей в обязательный набор знаний и умений всякого инженера, причастного к созданию и эксплуатации программного обеспечения компьютеров. Технология программирования имеет четко выделенный объект изучения – процессы разработки и сопровождения программного обеспечения, но, в настоящее время, не имеет единого метода и общепринятого способа построения. Технология программирования не является строгой математической дисциплиной, которую можно изложить последовательно, начиная с основополагающих понятий и применяя дедуктивные доказательства. Напротив, технология программирования является собранием разнородных и часто несогласованных друг с другом моделей, методик и средств. Кроме того, технологические приемы разработки программного обеспечения очень быстро меняются, почти каждые полгода предлагаются новые подходы.

В связи с этим сам предмет дисциплины – процесс разработки программного обеспечения – может трактоваться самым различным образом. Столь же дискуссионным является разделение технологии программирования на три составляющих: модель процесса, модель команды и дисциплину программирования.

Дискуссионные вопросы

1. Правомерно ли рассматривать предмет технологии программирования – процесс разработки программного обеспечения – как единое целое, или такое рассмотрение неправомерно, и процесс разработки необходимо дифференцировать и рассматривать только частные случаи?

2. Следует ли включать в технологию программирования вопросы организации взаимодействия людей, кодирования, тестирования, инструментальных средств и тому подобное, или же задачей технологии программирования является только описание процесса разработки, и более ничего?

Рекомендуемая позиция первого панелиста

1. Технология программирования имеет общее содержание, безотносительно к предметной области, платформе разработки, классу программных систем и так далее.

ЛП Системы представления знаний

2. Задачей технологии программирования является изучение всех аспектов и задач, связанных с разработкой программного обеспечения.

Рекомендуемая литература для подкрепления позиции:

1. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.
2. Орлов С. Технологии разработки программного обеспечения. – СПб.: Питер, 2002.

Рекомендуемая позиция второго панелиста

1. Класс программных систем, платформа разработки, корпоративные традиции и другие обстоятельства оказывают подавляющее влияние на процесс. В каждом отдельном случае действуют свои законы и следует применять свою технологию программирования.

2. Задачей технологии программирования является изучение только тех специфических аспектов и задач, которые реально возникают в каждом конкретном варианте процесса разработки программного обеспечения.

Рекомендуемая литература для подкрепления позиции:

1. Бек К. Экстремальное программирование. – СПб.: Питер, 2002.
2. Брукс-мл. Ф. П. Как проектируются и создаются программные комплексы. М.: Наука, 1975; новое издание перевода: Мифический человек-месяц. СПб.: СИМВОЛ+, 1999.
3. Терехов А.Н. Технология программирования. М.: БИНОМ, 2006.

Этапы жизненного цикла

Понятие жизненного цикла программы является ключевой абстракцией, вокруг которой строятся все известные в настоящее время технологические подходы к программированию.

Жизненный цикл программы – это некоторая абстрактная модель, однако элементы принимаемого жизненного цикла являются тем материалом, из которого состоят различные конкретные модели технологии программирования. Поэтому обсуждение понятия жизненного цикла программы необходимо должно предшествовать обсуждению элементов всякой технологии программирования.

Предложено несколько альтернативных моделей жизненного цикла, которые различаются, причем существенно, в различных аспектах.

Само понятие жизненного цикла и его составляющих представляется дискуссионным.

Дискуссионные вопросы

1. Можно ли считать, что все программы на самом верхнем уровне абстракции имеют один жизненный цикл, а отклонения от него являются не более чем «исключениями, подтверждающими правило»?

2. Целесообразно ли использовать единую номенклатуру составляющих жизненного цикла для всех моделей, и если да, то что следует включить в эту номенклатуру, а если нет, то как следует определять составляющие жизненного цикла?

Рекомендуемая позиция первого панелиста

1. Абстракция жизненного цикла для промышленного программирования имеет непреложный характер, а возможные отклонения от него регулируются промышленными стандартами.

2. Номенклатура, названия, содержание и порядок применения составляющих жизненного цикла определены отраслевыми, государственными и международными стандартами и не подлежат обсуждению.

Рекомендуемая литература для подкрепления позиции:

1. ГОСТ Р ИСО/МЭК 12207-99 «ПРОЦЕССЫ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММНЫХ СРЕДСТВ»

2. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.

Рекомендуемая позиция второго панелиста

1. Классические абстракции жизненного цикла полезны только как исходные идеи, которые могут пригодиться при конструировании модели каждого уникального процесса разработки.
2. Общая номенклатура составляющих невозможна, практики экстремального программирования не имеют ничего общего с процессами ГОСТ 12207.

Рекомендуемая литература для подкрепления позиции:

1. Бек К. Экстремальное программирование. – СПб.: Питер, 2002.

Сравнение моделей процесса разработки

Трудности разработки реальных приложений обусловлены их сложностью, и критическую роль в преодолении этой сложности играет сам процесс разработки.

Существует необозримое множество конкретных процессов разработки, практически в каждой программирующей организации, а иногда и в каждом проекте используется свой процесс разработки. Однако, как правило, все конкретные процессы следуют некоторым общим принципам и имеют много общего.

Абстрактное описание общих принципов, используемых в некотором классе сходных проектов, называется моделью процесса разработки.

Моделей процесса разработки предложено довольно много (хотя и меньше, чем конкретных процессов, разумеется). Разнообразные модели также можно сгруппировать и классифицировать в соответствии с теми основными допущениями, которые принимаются при конструировании модели. Выделенные таким образом классы моделей принято называть стратегиями конструирования моделей процесса разработки.

В рамках одной и той же стратегии и даже в рамках одной и той же модели процесс можно определить по-разному. Можно определить процесс максимально подробно и жестко, так чтобы не только были правила поведения на все случаи жизни, но правила нарушения правил, и правила изменения правил и так далее. Должны быть заранее определены типы артефактов, роли, периодическая отчетность и так далее. Только так определенный процесс обладает предсказательной силой, допускает измерение и управление. Такие процессы называют тяжеловесными. Но можно и не определять все заранее, а достраивать процесс по мере необходимости, на ходу, решать проблемы по мере их возникновения. Такие процессы называют гибкими, или облегченными.

Дискуссионные вопросы

Какой процесс выбрать – тяжеловесный или облегченный? Какие факторы влияют на этот выбор?

Рекомендуемая позиция первого панелиста

Процесс тем лучше, чем строже он регламентирован. В противном случае наступает анархия, потеря управляемости и другие неприятности. Строго регламентированный процесс всегда можно изменить в деталях и настроить на конкретный проект.

Рекомендуемая литература для подкрепления позиции:

1. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.
2. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб : Питер, 2002.

Рекомендуемая позиция второго панелиста

Регламентация бесполезна, она никогда не успевает за жизнью. Накладные расходы на организацию процесса никогда не окупаются. Чем легче процесс, тем он эффективнее.

Рекомендуемая литература для подкрепления позиции:

1. Бек К. Экстремальное программирование. – СПб.: Питер, 2002.
2. Терехов А.Н. Технология программирования. М.: БИНОМ, 2006.

Документированные процедуры

Основу описания процесса составляют документированные процедуры, используемые при проведении проекта, которые применяются к самому проекту, к бригаде проекта, к процессу проведения проекта, к материалам проекта и так далее.

Дискуссионные вопросы

Сколько должно быть документированных процедур и как они должны быть описаны?

Рекомендуемая позиция первого панелиста

Все действия, которые выполняются при проведении проекта по разработке программного обеспечения, должны подпадать под действие одной из документированных процедур. Все процедуры должны быть определены, документированы, утверждены и доведены до каждого исполнителя.

Рекомендуемая литература для подкрепления позиции:

1. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.
2. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб : Питер, 2002.
3. ГОСТ. ЕДИНАЯ СИСТЕМА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ (ЕСПД).

Рекомендуемая позиция второго панелиста

Документированные процедуры не нужны, достаточно нескольких общих принципов и здравого смысла, который есть у всех.

Рекомендуемая литература для подкрепления позиции:

1. Бек К. Экстремальное программирование. – СПб.: Питер, 2002.
2. Терехов А.Н. Технология программирования. М.: БИНОМ, 2006.

Сравнение моделей команды разработки

В современных условиях программное обеспечение разрабатывается коллективами, иногда очень большими коллективами. В таких коллективах проблемы взаимодействия являются постоянными и иногда очень сложными. Сами проблемы, несомненно, относятся к процессу разработки программного обеспечения, а способы решения этих проблем относятся к технологии программирования.

Дискуссионные вопросы

При каких масштабах производства программного обеспечения необходима регламентация производственных отношений?

Рекомендуемая позиция первого панелиста

Регламентация необходима, если в процесс вовлечено больше одного программиста.

Рекомендуемая литература для подкрепления позиции:

1. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.
2. Брукс-мл. Ф. П. Как проектируются и создаются программные комплексы. М.: Наука, 1975; новое издание перевода: Мифический человеко-месяц. СПб.: СИМВОЛ+, 1999.

Рекомендуемая позиция второго панелиста

Регламентация нужна, когда в проекте работают сотни и тысячи программистов. Пока команда помещается за одним столом, достаточно дружелюбия и природной общительности.

Рекомендуемая литература для подкрепления позиции:

1. Бек К. Экстремальное программирование. – СПб.: Питер, 2002.

Сравнение парадигм программирования

Вопрос повышения продуктивности программирования является предметом технологии программирования. Технология программирования находится в центре внимания ведущих практических программистов, теоретиков и начальников от программирования начиная с конца 60-х годов, со времен так называемой первой революции в программировании. Хотя ресурсы, инвестированные в технологию программирования, огромны и достижения значительны, вопрос далек от окончательного разрешения. Подтверждением тому служит разнообразие парадигм программирования, которые постоянно возникают, не вытесняя друг друга.

Дискуссионные вопросы

Какая парадигма программирования является наилучшей?

Рекомендуемая позиция первого панелиста

В современных условиях для промышленного программирования наилучшей является объектно-ориентированная парадигма. Все остальные парадигмы являются теоретически любопытными, но практически бесполезными академическими изысканиями.

Рекомендуемая литература для подкрепления позиции:

1. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.
2. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб : Питер, 2002.

Рекомендуемая позиция второго панелиста

Объектно-ориентированная парадигма является тормозом на пути дальнейшего прогресса программирования. Функциональное программирование (или логическое программирование, или автоматное программирование или другая парадигма по выбору панелиста) имеет неоспоримые преимущества.

Рекомендуемая литература для подкрепления позиции:

1. Материалы сайта <http://is.ifmo.ru>
2. Дал У., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир, 1975.

ЛП Системы представления знаний

3. Дейкстра Э. Дисциплина программирования. М.: Мир, 1978.
4. Хендерсон П. Функциональное программирование. Применение и реализация. М.: Мир, 1983.

Сравнение программных архитектур

Архитектура приложения — это набор принципов, используемых при принятии решений в процессе разработки и оценки приложения.

На архитектуру приложения (или программного решения иного типа) оказывают влияние множество факторов:

- тип приложения (информационная система, система управления, встроенная система и т.п.);
- характер использования (одноразовая, индивидуальная, групповая, массовая программа);
- дополнительные требования (надежность, секретность и пр.);
- предметная область (бизнес, наука, транспорт и т.п.);
- субъективные предпочтения пользователей и разработчиков и многое другое.

Сколько существует возможных комбинаций значений перечисленных (и не перечисленных) факторов, столько можно усмотреть различных (и в некотором смысле наилучших) архитектур. Таким образом, единственной рекомендуемой архитектуры (набора руководящих принципов) нет и быть не может.

Дискуссионные вопросы

Как выбрать программную архитектуру в конкретном проекте?

Рекомендуемая позиция первого панелиста

Следует знать все известные архитектуры и образцы проектирования, и исходя из требований к программному обеспечению, определить наиболее подходящие в данном случае и применить их.

Рекомендуемая литература для подкрепления позиции:

1. Брауде Э. Технология разработки программного обеспечения. – СПб. : Питер, 2004.
2. Орлов С. Технологии разработки программного обеспечения. – СПб.: Питер, 2002.

Рекомендуемая позиция второго панелиста

Архитектуру не следует определять в начале разработки, нужно просто по каждому архитектурному вопросу каждый раз принимать самое простое из возможных решений и в конце получится наилучшая для данного проекта архитектура.

Рекомендуемая литература для подкрепления позиции:

1. Бек К. Экстремальное программирование. – Спб.: Питер, 2002.
2. Терехов А.Н. Технология программирования. М.: БИНОМ, 2006.

Тенденции развития технологии программирования

Тема последней дискуссии должна выходить за рамки подготовленного учебно-методического пособия. В ходе этой дискуссии учащиеся должны обсудить новейшие тенденции в развитии технологии программирования, которые будут известны на момент проведения дискуссии. Конкретные позиции панелистов и рекомендуемые источники должен определить преподаватель, ведущий курс.

Дискуссионные вопросы

Каковы тенденции развития технологии программирования?
Какой будет технология программирования в ближайшем будущем?