

Технология разработки программного обеспечения

Lecture Notes N 8 v.[5]

Б. Мишнев



Технология разработки ПО

Система инженерных принципов для создания экономичного ПО, которое надежно и эффективно работает в реальных компьютерах

Этапы разработки ПО

- Планирование и оценка проекта
- Анализ системных и программных требований
- Проектирование алгоритмов, структур данных и программных структур
- Кодирование
- Тестирование
- Сопровождение

Первейшая цель проекта: высокое качество результата

- Высокое качество = своевременность выполнения проекта в соответствии с требованиями заказчика

Почему это выгодно всем?

- Меньше переделок и исправлений!

Процедуры разработки ПО обеспечивают:

- Порядок применения методов и утилит
- Формирование отчетов, форм
- Изменения и контроль качества
- Формирование «вех» для оценки прогресса руководством

Викторина 1

- Нарисуйте в конспекте таблицу

№ вопр.	Ответ	Проверка
1.1		
1.2		
1.3		
1.4		
1.5		

- В качестве ответов используйте:
I – да, - нет, O – не известно

Вопрос 1.1

- Правильно ли то, что умение строить новые программы сегодня отстает от требований к новым программам?

Вопрос 1.2

- Правильно ли то, что низкое качество программного обеспечения существенно снижает наши возможности по эксплуатации существующих программ?

Вопрос 1.3

- Верно ли, что международный стандарт по программной инженерии IEEE/ACM называется COSOMO II?

Вопрос 1.4

- Верно ли, что наиболее популярная модель для оценивания проектных затрат называется SWEVOK 2001 ?

Вопрос 1.5

- Верно ли, что штаб-квартира IEEE (Institute of Electrical and Electronic Engineers) находится в Лондоне?

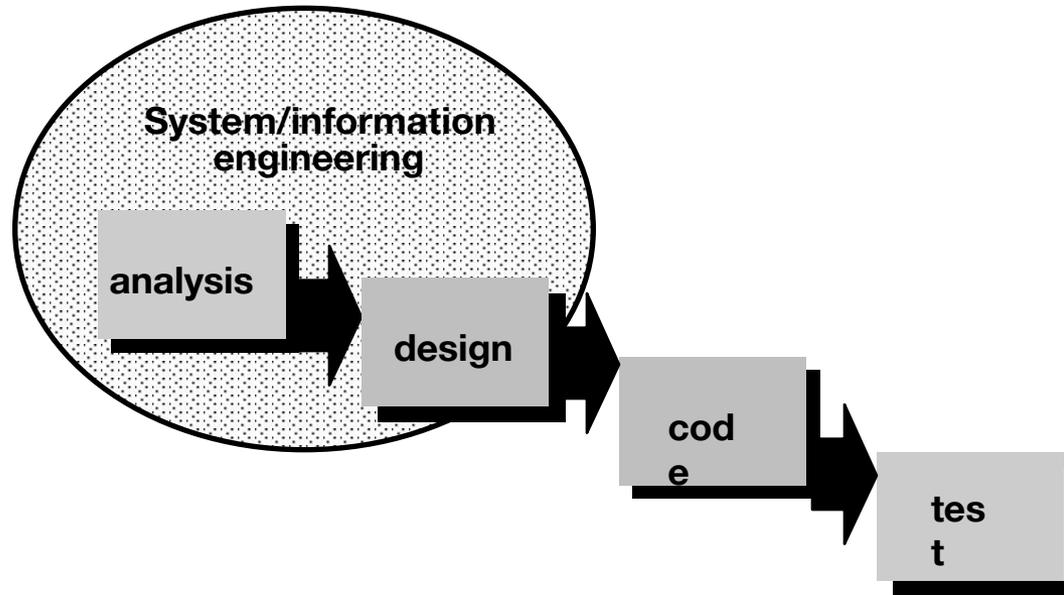
Викторина 1 (ответы)

- Сравните свои ответы и отметьте правильные в графе проверка «галочкой»

№ вопроса	Ответ	Проверка
1.1	I	
1.2	I	
1.3	-	
1.4	-	
1.5	-	

- Количество правильных ответов запишите!

«Водопадная модель» (Waterfall model)



Классический жизненный цикл ПО

- **Достоинства:**

- дает план и временной график по всем этапам проекта
- вносит дисциплину в ход разработки

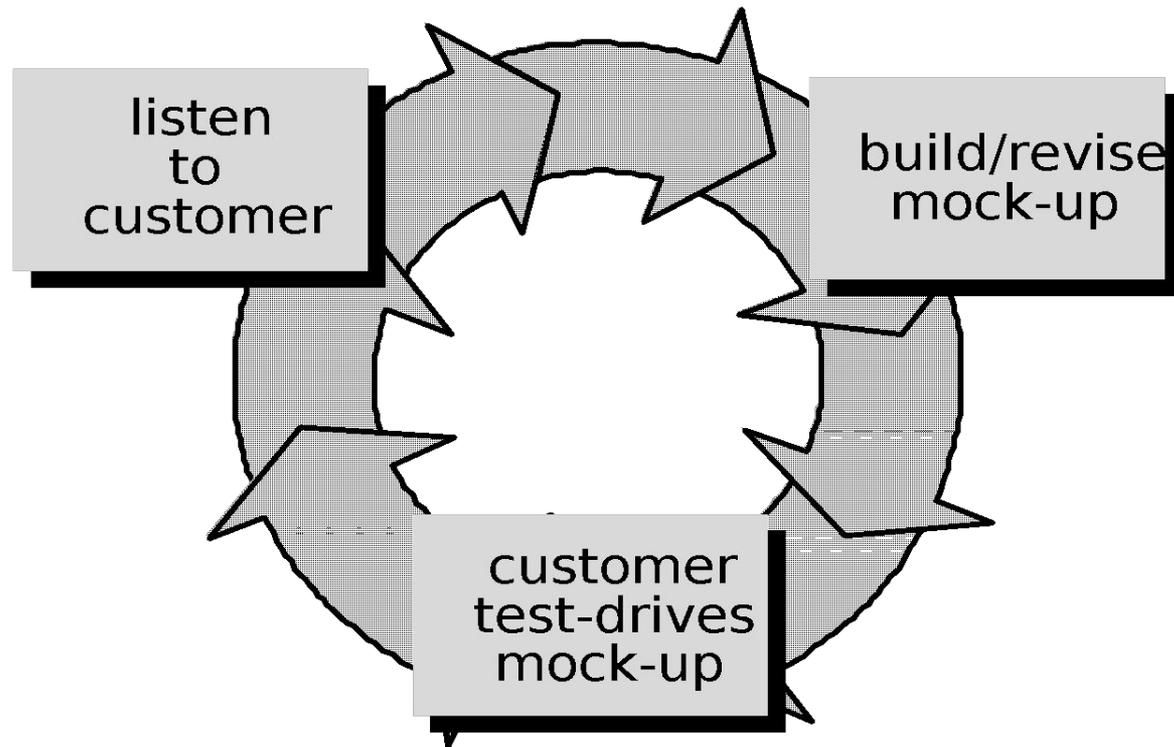
- **Недостатки:**

- Реальные проекты требуют отклонения от стандартной последовательности шагов
- Цикл основан на точной формулировке исходных требований к ПО
- Результаты доступны заказчику только в конце работы

Макетирование

- Процесс создания модели требуемого программного продукта
- Основная цель – снять неопределенность с требований заказчика

The Prototyping Models



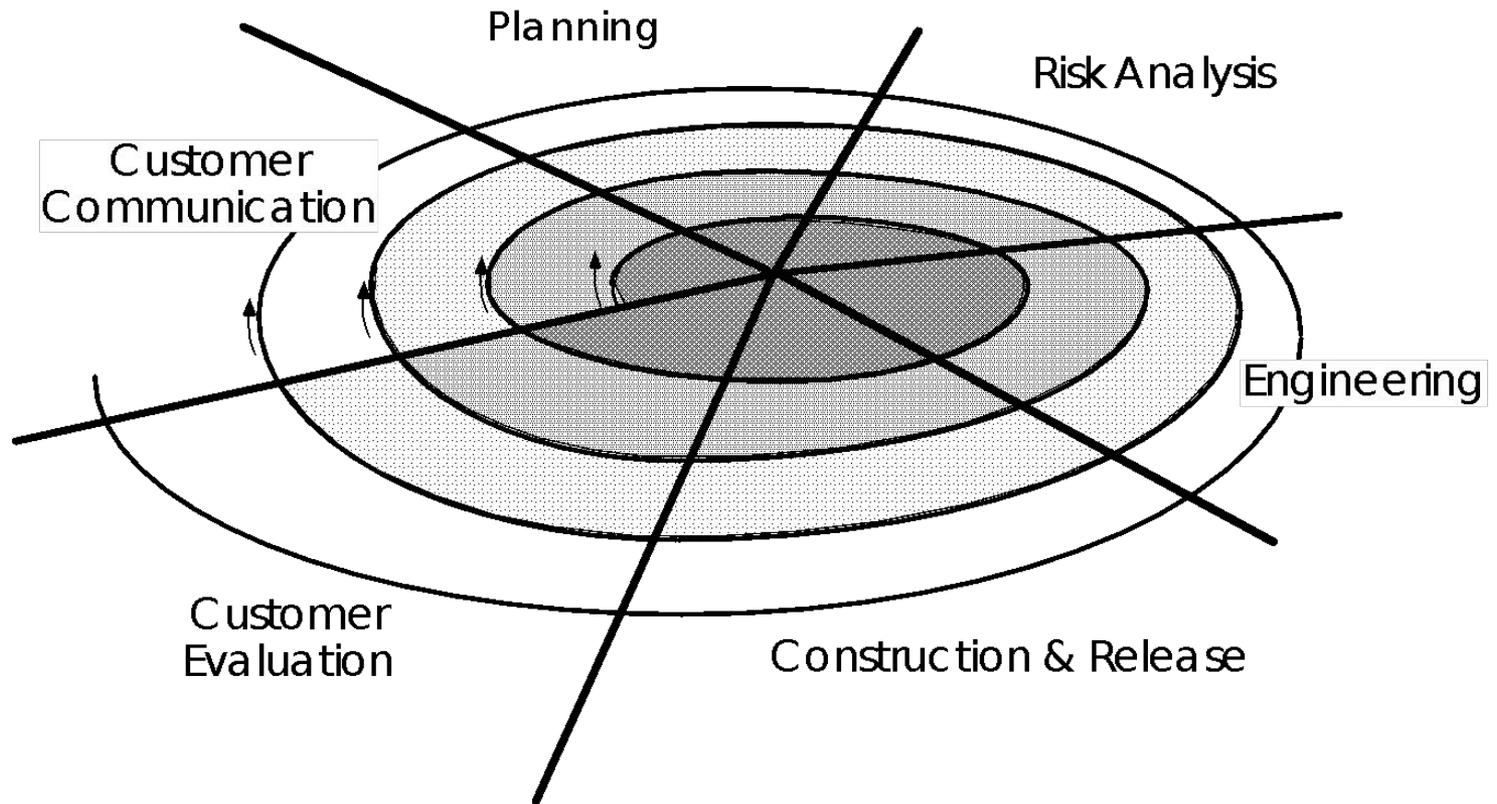
Формы моделей

- Статический макет – изображает человеко-машинный диалог (“screenshots”)
- Работающий макет
- Существующая программа, подлежащая улучшению

Спиральная модель (Бозм, 1988 – Воем, 1988)

- Планирование – определение целей, вариантов и ограничений
- Анализ риска – анализ вариантов и распознавание рисков
- Конструирование – разработка продукта следующего уровня
- Оценивание – оценка заказчиком текущих результатов конструирования

An Spiral Model



Спиральная модель

- Достоинства:
 - Реально отображает эволюцию ПО
 - Позволяет учитывать риски на каждом этапе эволюции разработки
 - Использует моделирование для снижения риска
- Недостатки:
 - Повышенные требования к заказчику
 - Трудности контроля и управления временем разработки

Викторина 2

- Нарисуйте в конспекте таблицу

№ вопр.	Ответ	Проверка
2.1		
2.2		
2.3		
2.4		
2.5		

- В качестве ответов используйте:
I – да, - нет, O – не известно

Вопрос 2.1

- Верно ли, что классический жизненный цикл ПО дает план и временной график по всем этапам проекта ?

Вопрос 2.2

- Правильно ли то, что классический жизненный цикл предполагает начало следующего этапа разработки только после завершения предыдущего?

Вопрос 2.3

- Верно ли, что зачастую проще разработать новую программу, чем успешно модифицировать существующую?

Вопрос 2.4

- Верно ли, что разработка программного продукта может быть успешна только тогда, когда с самого начала разработки все требования пользователя точно известны?

Вопрос 2.5

- Верно ли, что сопровождение (maintenance) – это внесение изменений в эксплуатируемое программное обеспечение?

Викторина 2 (ответы)

- Сравните свои ответы и отметьте правильные в графе проверка «галочкой»

№ вопроса	Ответ	Проверка
2.1	I	
2.2	I	
2.3	I	
2.4	-	
2.5	I	

- Количество правильных ответов запишите!

Методы проектирования

- **Нисходящие методы** – сложная задача сводится путем декомпозиции к набору более простых задач.
- **Восходящий методы** – проектирование системы начинается с определения отдельных задач внутри системы, которые затем используются для решения более сложных задач системы

Инструменты разработки

- Диаграммы потоков данных (Data Flow - DF)
- Диаграммы «сущность-связь» (Entity Relationship -ER)
- Словарь данных (Data Dictionary)
- Карты взаимодействия классов (Class Responsibility Collaboration – CRC)
- Язык моделирования (Unified Modeling Language – UML)

Шаблоны проектирования (Pattern Language by Alexander)

Предлагает парадигму архитектуры, основанную на трех понятиях: качестве, шлюзе и пути

- **Качество** – сущность, присущая всему живому и вещам, их польза
- **Шлюз** – механизм, который позволяет нам достигать качества
- **Путь** - позволяет применять образцы посредством шлюза

История – 1987 год

- В 1987 году Кент Бэк (**Kent Beck**) и Вард Каннигем (**Ward Cunningham**) взяли идеи **Кристофера Александра** и разработали шаблоны применительно к разработке программного обеспечения для разработки графических оболочек на языке **Smalltalk**.

История – 1991 год

- В этом году **Эрих Гамма** заканчивает свою докторскую работу и переезжает в США, где в сотрудничестве с Ричардом Хелмом (Richard Helm), Ральфом Джонсоном (Ralph Johnson) и Джоном Влиссидсом (John Vlissides) публикует книгу **Design Patterns — Elements of Reusable Object-Oriented Software**.

«Банда четырех»

- В этой книге описаны 23 шаблона проектирования.
- Также команда авторов этой книги известна общественности под названием **Gang of Four** (GoF).
- Именно эта книга послужила причиной широкого распространения метода шаблонов.

Software design patterns

A. Creational

- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton

B. Structural

- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

C. Behavioral

- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

Викторина 3

- Нарисуйте в конспекте таблицу

№ вопр.	Ответ	Проверка
3.1		
3.2		
3.3		
3.4		
3.5		

- В качестве ответов используйте:
I – да, - нет, O – не известно

Вопрос 3.1

- Правильно ли то, что при нисходящих методах проектирования возможны неразрешимые проблемы для модулей нижнего уровня?

Вопрос 3.2

- Правильно ли то, что при восходящих методах проектирования гарантированно достигается выполнение всех требований пользователя на высшем уровне?

Вопрос 3.3

- Верно ли, что ER диаграммы эффективно используются при проектировании структур баз данных?

Вопрос 3.4

- Верно ли, что паттерн или образец – это "некий общий принцип определения задачи, повторно встречающийся в окружающей среде и фиксирующей контекст»?

Вопрос 3.5

- Верно ли, что книгу «Паттерны проектирования» написала «банда четырех» под руководством Эрика Гаммы?

Викторина 3 (ответы)

- Сравните свои ответы и отметьте правильные в графе проверка «галочкой»

№ вопроса	Ответ	Проверка
3.1	I	
3.2	-	
3.3	I	
3.4	I	
3.5	I	

- Количество правильных ответов запишите!

Тестирование (verification)

Процесс выполнения программы с целью обнаружение существующих ошибок

Тест определяет:

- Свой набор исходных данных и условия для запуска программы
- Набор ожидаемых результатов

Принципы тестирования

- **Функциональное тестирование** (тестирование «черного ящика») – выполняется с точки зрения пользователя системы
- **Структурное тестирование** (тестирование «белого ящика») – основывается на знании внутренней структуры программы

Документирование

- Документация проекта (требования к системе, план работ, тест-план, отчеты и .т.д.)
- Документация продукта
 - Пользовательская (Руководство пользователя, Инструкции по установке и .т.д.)
 - Системная (текст программ, описание программных интерфейсов и т.д.)

Метрики

- **Размерно-ориентированные метрики (основываются на Lines Of Code – LOC)**
 - Широко распространены
 - Легко вычисляются
 - Зависят от языка программирования
- **Функционально-ориентированные метрики (основываются на специально рассчитанных функциональных указателях – Function Points – FP)**
 - Не зависят от языка программирования
 - Основаны на субъективных данных и используют косвенные измерения

Управление проектом

Планируются и контролируются следующие параметры:

- Потребные человеческие ресурсы (**efforts**) – чел. месяцы / дни
- Время выполнения проекта (**time**) – календарные месяцы / дни
- Размер программного продукта (**size**) – KLOC / FP.
- Требуемое качество продукта (**quality**)

IBM Rational Unified Process

- Новый подход к разработке ПС, основанный на использовании лучших практических методов, успешно зарекомендовавших себя во многих проектах разработки ПС по всему миру;
- Четко определенный процесс (технологическая процедура), описывающий структуру жизненного цикла проекта, роли и ответственности отдельных исполнителей, выполняемые ими задачи и используемые в процессе разработки модели, отчеты и т.д.;
- Готовый продукт, предоставляемый в виде веб-сайта, содержащего все необходимые модели и документы с описанием процесса.

RUP



Инструментальные средства IBM Rational

- **Управление требованиями** – IBM Rational RequisitePro;
- **Визуальное моделирование и генерация объектного кода** – IBM Rational Rose, IBM Rational XDE;
- **Разработка** - IBM Rational RapidDeveloper
- **Конфигурационное управление** – IBM Rational ClearCase;
- **Управление изменениями** – IBM Rational ClearQuest;
- **Автоматизированное документирование** – IBM Rational SoDA;
- **Автоматизированное тестирование** – IBM Rational TeamTest, IBM Rational TestFactory, IBM Rational Robot, IBM Rational PurifyPlus, IBM Rational SiteCheck и IBM Rational SiteLoad.

Викторина 4

- Нарисуйте в конспекте таблицу

№ вопр.	Ответ	Проверка
4.1		
4.2		
4.3		
4.4		
4.5		

- В качестве ответов используйте:
I – да, - нет, O – не известно

Вопрос 1.1

- Правильно ли то, что с помощью тестирования по принципу «черного ящика» можно обнаружить некорректные или отсутствующие в продукте функции?

Вопрос 1.2

- Правильно ли то, что с помощью способов тестирования, основанных на принципе «белого ящика», можно получить оценку комплексной сложности программы?

Вопрос 1.3

- Верно ли, что пользовательская документация должна разрабатываться на самом заключительном этапе проекта?

Вопрос 1.4

- Верно ли, что размерно-ориентированные метрики не зависят от языка программирования?

Вопрос 1.5

- Верно ли, что в RUP входят 6 основных дисциплин:
 - Бизнес-моделирование (Business modeling);
 - Управление требованиями (Requirements);
 - Анализ и Проектирование (Analysis and Design);
 - Реализация (Implementation);
 - Тестирование (Test);
 - Развертывание (Deployment)?

Викторина 4 (ответы)

- Сравните свои ответы и отметьте правильные в графе проверка «галочкой»

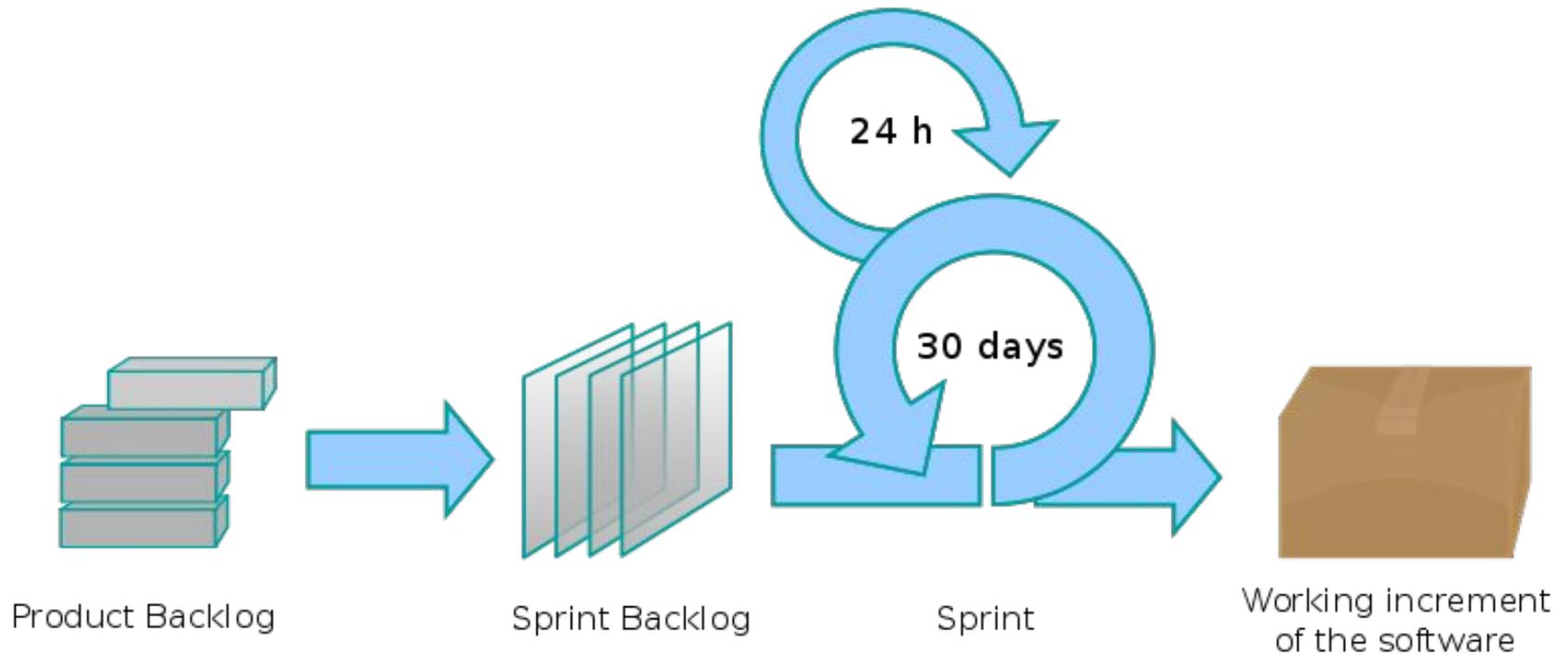
№ вопроса	Ответ	Проверка
4.1	I	
4.2	I	
4.3	-	
4.4	-	
4.5	-	

- Количество правильных ответов запишите!

Scrum

- Scrum — методология управления разработкой информационных систем для гибкой разработки программного обеспечения. Scrum чётко делает акцент на качественном контроле процесса разработки.

Scrum



SW-CMM (CMMI)

- **Capability Maturity Model (CMM)**

- Levels of the CMM**

- Level 1 - Initial

- Level 2 - Repeatable

- Level 3 - Defined

- Level 4 - Managed

- Level 5 - Optimizing

Литература

- Дж. Г. Брукшир. Введение в компьютерные науки. – М., «Вильямс», 2001, с.341-378.
- С.А. Орлов. Технологии разработки программного обеспечения.- СПб.: Питер, 2002. – 464 с.
- Майк Кон. Scrum: гибкая разработка ПО = Succeeding with Agile: Software Development Using Scrum — М.: «Вильямс», 2011. — 576 с.

Спасибо за внимание!

Dr. Sc Ing. Борис Мишнев

