

Технология программирования



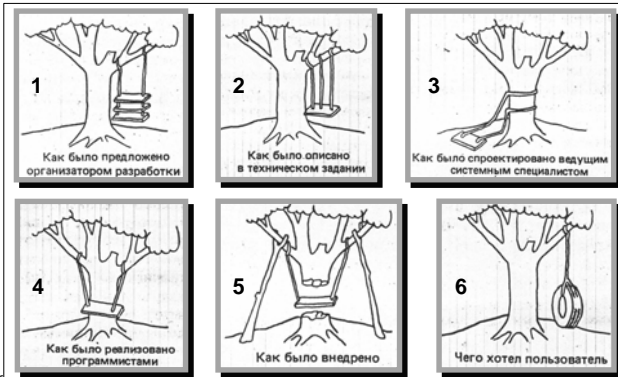
Начинать важное дело без серьезной подготовки есть первый признак шизофрении.

Бехтерев

Компьютерные науки

© М.Л. Цымблер

Эпиграф



Технология программирования

© М.Л. Цымблер

2

Содержание

- Основная проблема программирования
- Понятие технологии программирования
- Жизненный цикл разработки программного обеспечения

Технология программирования

© М.Л. Цымблер

3

Основная проблема программирования

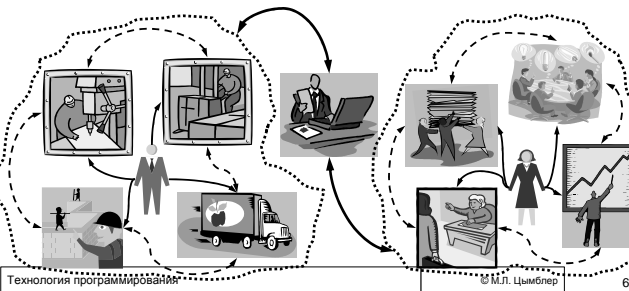
- Программное обеспечение, как правило, имеет
 - большое время жизни,
 - большое количество пользователей, зависящих от его нормального функционирования,
 - большую команду разработчиков (один разработчик не в состоянии охватить все аспекты системы).
- Основной проблемой разработки программного обеспечения является **СЛОЖНОСТЬ**.
Сложность неизбежна: с ней можно справиться, но избавиться от нее нельзя.

Основные причины сложности

- Сложность предметной области, из которой исходит заказ на разработку ПО.
- Сложность управления процессом разработки.

Внутренняя сложность предметной области

- Огромное количество элементарных действий над огромным количеством элементарных объектов с огромным количеством связей между ними.





Внешняя сложность предметной области

- 
 Пользователю **сложно** объяснить разработчикам, *что* должна делать система (часто он сам этого не знает).
- 
 Разработчикам **сложно однозначно** зафиксировать все требования пользователя.
- 
 Дополнительная **сложность** из-за изменения требований пользователя к системе *в процессе* разработки.

Технология программирования © М.Л. Цымблер 7

Сложность управления процессом разработки

- 
Невозможность полного понимания *одним* человеком программной системы объемом в десятки-сотни тысяч строк (сотни-тысячи модулей).
- 
Сложность координации между разработчиками, особенно в случае их географической удаленности друг от друга.

Технология программирования © М.Л. Цымблер 8

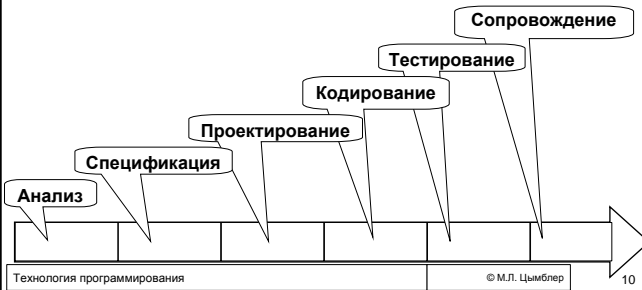
Технология программирования

- Технология программирования (*Software Engineering*)** – совокупность теории и практической техники, на которые опирается процесс создания программного обеспечения.
- Технология программирования включает в себя
 - концептуальные средства**, определяющие стиль и методы разработки программ
 - организационные средства**, определяющие форму труда в команде программистов
 - программные средства** разработки.

Технология программирования © М.Л. Цымблер 9

Жизненный цикл ПО

■ *Жизненный цикл ПО (software lifecycle)* – период его разработки и эксплуатации, начиная от момента возникновения замысла ПО и заканчивая прекращением всех видов его использования.

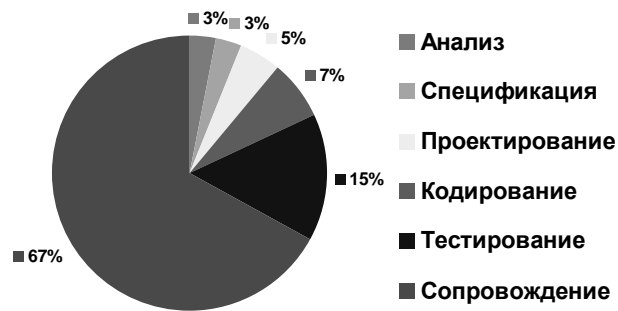


Технология программирования

© М.Л. Цымблер

10

Длительность этапов жизненного цикла



Технология программирования

© М.Л. Цымблер

11

Анализ



Прежде чем решать задачу, полезно познакомиться с ее условием.

Д. Поля

Технология программирования

© М.Л. Цымблер

12

Анализ требований

- На этапе *анализа требований (analysis)* разработчик *совместно* с пользователем устанавливают, *что* должна делать программная система (но не *как* она это должна делать).
- Установленные на этапе анализа требования к системе будут зафиксированы на этапе *спецификации*.

Технология программирования

© М.Л. Цымблер

13

Проблемы этапа анализа



- Трудность нахождения общего языка между программистом и пользователем (каждый из них не является специалистом в «чужой» области).



- Отсутствие у пользователя четкого представления о том, какая система ему нужна.

Технология программирования

© М.Л. Цымблер

14

Спецификация



*Все, что вообще может быть сказано,
должно быть сказано ясно,
а о чем невозможно говорить,
о том следует молчать.*

Л. Витгенштейн

Технология программирования

© М.Л. Цымблер

15

Спецификация требований

- На этапе *спецификации (specification)* требования пользователя записываются на достаточно формальном языке.
- *Спецификация* – документ, в котором точно (формально, однозначно), понятно (ясно) и полно записано, *что* должна делать программная система.
- Спецификация является единственным документом, по которому определяется успех (правильность) конечных результатов разработки.
- Спецификация должна быть понятна пользователю и программисту.

Технология программирования

© М.Л. Цымблер

16

Методы спецификации требований

- *Неформальная спецификация* использует только естественный язык.
- *Формальная спецификация* использует только строгую (математическую или др.) нотацию.
- *Полуформальная спецификация* использует комбинацию естественного языка с различными диаграмматическими и табличными нотациями.

Технология программирования

© М.Л. Цымблер

17

Примеры спецификации

1. Возьми само число
И те, что перед ним,
Всё перемножь; тогда
Получишь, что искал.
2. $f \equiv \text{eq} \circ [\text{id}, 1] \rightarrow 1;$
 $x \circ [\text{id}, f \circ - \circ [\text{id}, 1]]$
3. $F(n, n*y) :- F(m, y), m = n-1$
 $F(1, 1)$
4. $f(n) = \text{if } n \leq 1 \text{ then } 1$
 $\text{else } n * f(n-1)$
5. $f(n) = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n$

```
function Factorial
  (N: Integer): Integer;
var
  i, F: Integer;
begin
  F:=1;
  for i:=1 to N do
    F:=F*i;
  Factorial:=F;
end;
```

Технология программирования

© М.Л. Цымблер

18

Проблемы этапа спецификации



- Выбор метода спецификации.



- Сложность итогового документа спецификации.

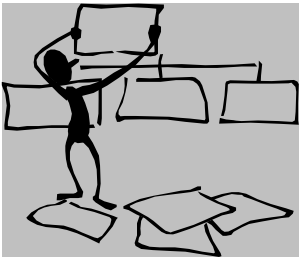


- Невозможность предусмотреть в спецификации абсолютно всё.



- Высокая цена возможных ошибок в спецификации.

Проектирование



Зри в корень!
Козьма Прутков

Разделяй и властвуй!
Макиавелли

Проектирование системы

- На этапе *проектирования (design)* разрабатывается модульная структура системы:

1. Система (основная задача) представляется в виде иерархии подсистем (подзадач).
2. Каждая подсистема представляется в виде иерархии *модулей* (**автономно** разрабатываемых подпрограмм).

Для каждого модуля выполняется *спецификация*: назначение модуля, формальные параметры (входные и выходные данные), возвращаемый результат и его семантика.

Модульная структура системы

```

function Calc(S: String; var
Error: Integer): Real;
{ Вычисление значения
выражения.
S – строковая запись выражения.
Error – тип ошибки в выражении:
  0 – нет ошибок
  -1 – синтаксическая
  -2 – арифметическая
В случае арифметической или
синтаксической ошибки в
выражении возвращает ноль. }
    
```

Технология программирования © М.Л. Цымблер 22

Спецификация модуля

- Спецификация модуля (подпрограммы) должна содержать:
 1. Назначение модуля
 2. Описание формальных параметров модуля и их семантики
 3. Описание возвращаемого результата и его семантики (для функций)
 4. Описание возможных побочных эффектов.

Технология программирования © М.Л. Цымблер 23

Проблемы этапа проектирования

- Для одной и той же (под)системы возможно предложить более одной схемы модульной структуры:
 - по какому принципу объединять операторы в модуль?
 - насколько модуль должен быть зависим или независим от других модулей?
 - каким должен быть размер модуля?

Технология программирования © М.Л. Цымблер 24

Кодирование



Стремитесь к простоте, насколько это возможно, но не более того.

А. Эйнштейн




Технология программирования © М.Л. Цымблер 25

Кодирование системы

- На этапе *кодирования (coding)* разработчик
 1. разрабатывает алгоритмы
 2. выбирает язык(и) программирования
 3. создает тексты модулей в виде файлов.

Технология программирования © М.Л. Цымблер 26

Проблемы этапа кодирования

- 
 ■ Выбор языка программирования и адекватной системы программирования.
- 
 ■ Разработка алгоритмов. Выбор (самого простого) варианта реализации алгоритма.
- 
 ■ Отсутствие «лесенки», спецификаций модулей, комментариев (не только у программистов-студентов).

Технология программирования © М.Л. Цымблер 27

Тестирование



*Доверяй, но проверяй.
Русская поговорка*

Тестирование системы

- На этапе *тестирования (testing)* разработчик выполняет программные тексты системы на *заранее подготовленном им наборе данных (тесте)*, для которого известен результат работы или поведение системы.
- **Цель тестирования** – не доказать правильность системы (это невозможно!), а продемонстрировать факт наличия в ней ошибки.

Автономное и комплексное тестирование

- *Автономное тестирование* – тестирование каждого модуля (под)системы в отдельности (без связи с другими модулями).
- *Комплексное тестирование* – тестирование взаимодействия всех модулей (под)системы.

Подготовка тестов

- Тесты готовятся на этапе проектирования. После того, как определен интерфейс модуля, разрабатывается тестовая программа и подбираются тестовые данные.
- Тестовые данные должны проверить работу модуля в обычных и граничных условиях.
- Тест должен с высокой вероятностью обнаруживать ошибку, а не демонстрировать правильную работу модуля.

Проведение тестов

- Проведение тестов нужно документировать.

№	Входные данные	Ожидаемый результат		Действительный результат		Тест пройден?
		Error	Calc	Error	Calc	
1	'2*2'	0	4.0	0	4.0	√
2	'3-'	-1	0.0	-2	0.0	×
3	'5/(2-2)'	-2	0.0	-2	0.0	√
...
N	"	-1	0.0	0	0.0	×

Проблемы этапа тестирования



- Разрабатывать тесты для модуля на основе его спецификации или на основе его реализации?



- Сколько должно быть тестов?



- Когда можно завершить тестирование?

Отладка

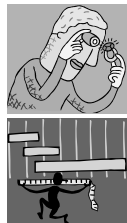


*Лишь та – ошибка,
что не исправляется.
Конфуций*

Отладка системы

- Во время *отладки (debugging)* локализуются и исправляются ошибки, обнаруженные во время тестирования.
- При отладке необходимо сочетать использование отладчика и «сухую» отладку.
- Необходимо автономное тестирование отлаженного модуля и комплексное тестирование всех подсистем, сопряженных с ним. В случае обнаружения неполноты набора тестовых данных нужно добавить в него соответствующие тесты.

Проблемы отладки



- Что считать ошибкой?
- Как локализовать и исправить ошибку?
- Как оценить общее количество ошибок в системе?

Сопровождение



*Мы в ответе за тех,
кого приручили.
А. де Сент-Экзюпери*

Сопровождение системы

- На этапе *сопровождения (maintenance)* осуществляется устранение ошибок, обнаруженных в системе во время эксплуатации, а также доработка и модификация системы вследствие изменения нужд пользователей.
- Сопровождение – неотъемлемый и самый долговременный этап жизненного цикла.

Проблемы этапа сопровождения

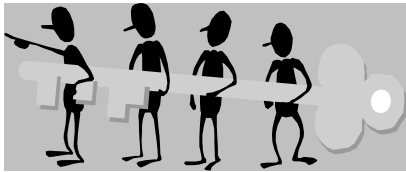


- Сопровождение системы для программиста психологически более трудно, чем собственно разработка системы.



- *Версионирование* исходных текстов системы: сохранение истории изменений с возможностью их отката до любого шага, возможность распространения изменений, вносимых в раннюю версию системы, на более поздние версии.

Коллективная разработка



*Когда в товарищах согласия нет,
на лад их дело не пойдет.*

И. Крылов

Проблемы коллективной разработки



- Какие роли должны быть в команде и как их распределить?
- Какие концептуальные и программные средства выбрать в качестве стандартных в команде?
- Как повысить мотивацию участников проекта?
- Что делать, если проект не укладывается в сроки?
