

**Министерство образования и науки Российской Федерации
ФГАОУ ВО «Севастопольский государственный университет»**

**Институт информационных технологий
и управления в технических системах
Кафедра «Информационные системы»**

СИСТЕМНАЯ ИНЖЕНЕРИЯ

Учебно-методическое пособие

II семестр

для студентов всех форм обучения направления подготовки
09.04.02 «Информационные системы и технологии» (магистратура)



Севастополь
2017

УДК 004.07

Учебно-методическое пособие по дисциплине «Системная инженерия» для студентов всех форм обучения направления подготовки 09.04.02 «Информационные системы и технологии» (магистратура) / Разраб. Ю.В. Доронина, В.Ю. Карлусов, И.В. Дымченко – Севастополь: Изд-во СевГУ, 2017. – xx с.

Учебно-методическое пособие рассмотрено и утверждено на заседании кафедры «Информационные системы», протокол № ____ от «31» августа 2016 г.

Допущено учебно-методическим центром СГУ в качестве учебно-методического пособия.

Содержание

Введение	4
Лабораторный практикум	4
1.1 ЛАБОРАТОРНАЯ РАБОТА № 1	4
ИССЛЕДОВАНИЕ АСПЕКТОВ НАДЁЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СИСТЕМНОЙ ИНЖЕНЕРИИ	4
ЛАБОРАТОРНАЯ РАБОТА № 2	14
ИССЛЕДОВАНИЕ ТЕХНИЧЕСКИХ АСПЕКТОВ ПРОЕКТИРОВАНИЯ В СИСТЕМНОЙ ИНЖЕНЕРИИ	14
ЛАБОРАТОРНАЯ РАБОТА № 3	17
ИССЛЕДОВАНИЕ АСПЕКТОВ КОЛИЧЕСТВЕННОГО АНАЛИЗА КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ В СИСТЕМНОЙ ИНЖЕНЕРИИ	17
ЛАБОРАТОРНАЯ РАБОТА № 4	21
ИССЛЕДОВАНИЕ ЭКОНОМИЧЕСКИХ АСПЕКТОВ АНАЛИЗА КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ В СИСТЕМНОЙ ИНЖЕНЕРИИ	21
ЛАБОРАТОРНАЯ РАБОТА №5 ФОРМУЛИРОВКА И АНАЛИЗ ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ	24
ЛАБОРАТОРНАЯ РАБОТА № 6	28
АНАЛИЗ ПОДХОДОВ К ФОРМАЛИЗАЦИИ ОПИСАНИЯ СЛОЖНЫХ СИСТЕМ	28
Требования к содержанию и оформлению отчетов	38
Организация защиты и критерии оценивания выполнения	39
лабораторных работ	39
Список экзаменационных вопросов	42
Приложение 1	44
Образец оформления и содержания отчета по лабораторной работе	44

Введение

Учебное пособие охватывает два семестра лабораторного практикума и курсовое проектирование, а также некоторые теоретические сведения и библиографию по дисциплине «Системная инженерия».

Лабораторные занятия обеспечивают подготовку студентов к выполнению курсового проекта, а также поддерживают активизацию исследовательской деятельности магистрантов в рамках подготовки выпускной квалификационной работы (ВКР) с использованием требуемого математического аппарата для анализа и моделирования сложных систем различной физической природы.

Лабораторный практикум

Цель проведения лабораторных занятий: закрепление теоретических знаний, полученных на лекционных занятиях и привитие навыков самостоятельной работы с различными аспектами системной инженерии.

1.1 ЛАБОРАТОРНАЯ РАБОТА № 1 ИССЛЕДОВАНИЕ АСПЕКТОВ НАДЁЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СИСТЕМНОЙ ИНЖЕНЕРИИ

1. Цель работы:

- Получить практические навыки в построении моделей отказов технических и программных средств в среде MatLab Simulink.
- Оценить надёжность программного обеспечения (ПО) с использованием моделей Нельсона и Коркорена.

2. Теоретические сведения

Базовой составной частью информационных систем (ИС) является ПО, в связи с чем изучение подходов для создания качественного ПО представляется важным для системно инженерного взгляда на ИС.

2.1. Расчётные выражения

Успешное решение задачи создания надёжного ПО позволяет избежать негативных последствий и затратных модификаций на этапе эксплуатации. По определению [1], программа обладает свойством надёжности в той мере, в которой она способна удовлетворять выполнению функции, для которых она предназначена. Под “удовлетворительно” понимается отсутствие изъянов, для устранения которых необходимо внести изменения в тексты программных

модулей.

Для точного определения надёжности в математическом смысле, используют термины статистики. В частности, Нельсоном предложено следующее[2].

1. Программа представляется как некоторая вычисляемая функция F на множестве E наборов входных данных. Элемент этого множества E_i представляет собой совокупность значений данных, необходимых для **однократного** выполнения программы.

$$E = \{E_i : i = 1, 2, \dots, N\}.$$

2. Совокупность действий, включающая ввод комбинации данных E_i , выполнение программы, завершаемое получением результата $F(E_i)$, называется **прогоном программы** или просто **прогоном**.

3. В ходе выполнения прогона происходит определение значения функции $F(E_i)$ для каждого набора E_i . Наличие отклонений от спецификации, вольных или невольных, приводит к отклонению $F(E_i)$ от требуемого. Иногда это отклонение находится в допустимых пределах

$$|F^V(E_i) - F(E_i)| \leq \delta_i$$

и остаётся незамеченным. В противном случае результат вычисления признаётся неприемлемым, и фиксируется так называемый “**рабочий отказ**”.

В результате последовательности прогонов может быть образовано (определено) множество наборов E_{Er} данных, приводящих к рабочим отказам.

Таким образом, вероятность безотказного функционирования (надёжности) ПО составит

$$R = 1 - P_{отк} = 1 - \frac{N_{Er}}{N}, \quad (1)$$

где N и N_{Er} – мощности множеств E и E_{Er} соответственно. Формула (1) верна в том случае, когда выбор набора входных данных осуществляется равновероятно. Так же использованию (1) препятствует невозможность исчерпывающего тестирования ПО.

На практике, выбор определённого теста определяется режимами работы ПО, которые необходимо проверить и связан с **распределением вероятностей** $P_i = P(E_i)$, множество которых называется **функциональным разрезом**. В этом случае, вероятность того, что прогон программы на множестве входных данных E закончится отказом, составит

$$P_{отк} = \sum_{i=1}^N q_i \cdot p_i,$$

где $q_i = \{0, 1\}$ – “динамическая” переменная, единичное значение которой соответствует рабочему отказу. Отсюда, вероятность безотказной работы

$$R = 1 - P_{отк} = \sum_{i=1}^N q_i \cdot (1 - p_i)$$

Поскольку в каждом прогоне используется свой функциональный разрез, а число прогонов ограничено, надёжность ПО будет определяться вероятностью того, что ни один из последовательности из n прогонов на различных функциональных разрезах не закончится отказом

$$R_n = \prod_{j=1}^n (1 - P_{OTK(j)}) = e^{\sum_{j=1}^n \ln(1 - P_{OTK(j)})} = e^{-\sum_{j=1}^n P_{OTK(j)}}, \quad (2)$$

где j – номер прогона, а $P_{OTK(j)}$ – вероятность отказа на этом прогоне.

Модель Нельсона использует гипотезу о том, что ошибки фиксируются в ходе прогонов программы, но не исправляются.

Для оценки надёжности функционирования ПО также применяется модель Коркорэна [3], которая учитывает динамику его отладки:

$$R_n = \frac{1}{n} \left(n_0 + \sum_{j=1}^T k_j (n_j - 1) \right), \quad (3)$$

где n_0 – число успешных испытаний в серии из n прогонов программы; T – число типов ошибок, по которым ведётся учёт; n_j – число ошибок j -го типа, выявленных за n прогонов; k_j – масштабирующий коэффициент, определяемый ступенчатой функцией

$$k_j = \begin{cases} q_j, & n_j > 0, \\ 0, & n_j = 0 \end{cases} \quad (4)$$

где q_j – вероятность устранения ошибки j -го типа.

2.2. Принципы моделирования процесса отладки ПО

С точки зрения теории вероятностей, обнаружение ошибки в процессе прогона является составным событием, комбинирующим два элементарных события:

- выбор из множества E набора E_i входных данных в соответствии с функциональным разрезом исследуемой программы;
- проявление рабочего отказа на выбранном наборе.

Поэтому для имитационного моделирования понадобятся два независимых источника (генератора) псевдослучайных чисел с различными законами распределения: распределение 1, имитирующее функциональный разрез, и распределение 2, моделирующее появление отказа. Псевдослучайные числа сравниваются между собой, и, в зависимости от результатов сравнения, генерируется сигнал рабочего отказа.

Схема моделирования представлена на рисунке 1 ниже. Остановимся на описании блоков MatLab Simulink, входящих в её состав [5].

Генерацию псевдослучайных чисел будем осуществлять с помощью блоков MatLab Simulink разделов конструктора Library: Sources/Constant и Library: NonLinear/MatLab function. Блоки в рабочее поле модели помещаются “перетаскиванием” из соответствующих разделов библиотеки конструкций.

Двойной щелчок на элементе открывает вкладку его свойств. На вкладке свойств константы есть поле для приёма её значения (которое после ввода отобразится на схеме). На вкладке MatLab function – поле для приёма

арифметического выражения, которое может иметь довольно сложную структуру. Для блока Raspred1 использована запись вида `exprnd(u,1,1)`, в которой обозначено: `exprnd` – функция генерирующая ряд чисел, отвечающих экспоненциальному распределению, `u` – принятое в стандарте MatLab Simulink обозначения входного сигнала блока, а две единицы – размер генерируемой матрицы (поскольку нас интересует одно число за цикл моделирования).

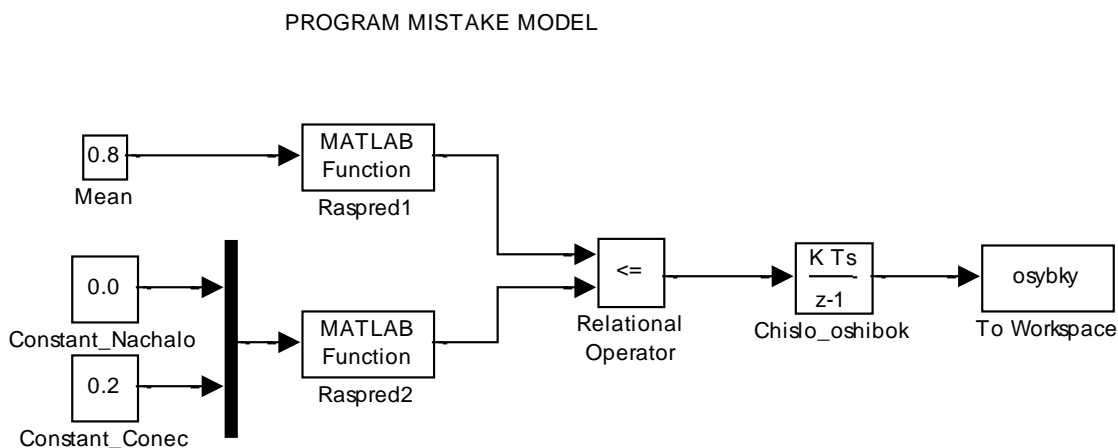


Рисунок 1 – Имитационная модель процесса тестирования

Для блока Raspred2 – `unifrnd(u(1),u(2),1,1)`, где – функция генерирующая равномерно распределенные значения в интервале от `u(1)` до `u(2)`. Чтобы сгруппировать входные значения блока в вектор используется элемент Library: Connection/Mux, имеющий единственный параметр – число входов. Порядковый номер элемента в векторе определяется номером входа Mux. Поскольку сам по себе элемент Mux малоинформативен, его графическое изображение “ужимают” манипуляторами до размера жирной линии. При этом название блока пропадает.

Блок раздела конструктора моделей Library: NonLinear / Relational Operator. Проверяет отношение, указанное на блоке, и, в случае его выполнения, генерирует сигнал “1”, в противном случае “0”. Отношение может быть изменено на любое из известных в математике, на вкладке свойств блока в выпадающем списке. Изменение отразится на изображении блока.

Блок Library: Discrete / Discrete-Time Integrator используется для подсчёта единичных импульсов, поступающих от блока сравнения. В блоке выполняется интегрирование по методу прямоугольников, и, тем самым определяется число ошибок.

Блок Library: Sinks / To Workspace используется для связи с программой на внутреннем языке MatLab, с целью последующей обработки. Связь осуществляется по идентификатору, задаваемом на вкладке свойств и отображаемом на изображении блока, в нашем случае употреблено имя `osybky`.

Имитационная модель может запускаться как в “ручном”, так и в “автоматическом” режимах. Перед началом запуска, в любом случае, на вкладке меню Simulations / Parameters задайте значения полей Stop Time, в поле Type установите значение Fixed-step, а в поле рядом discrete (no continuous states).

В поле Stop Time можно установить значение 100, которое достаточно для набора статистики по математическому ожиданию и удобно для расчётов.

Для однократного “автоматического” запуска модели необходимо использовать конструкцию MatLab `sim('PO_imit')`, и чтобы файл с моделью находился в одной директории с программой, осуществляющей запуск и обработку.

Доступ к выходным данным модели производится по ссылке, в нашем случае это `osybyk.signals.values`.

Каждый из запусков модели интерпретируется как испытание по одной из функций разреза Stop Time наборами данных.

В частности, для экспоненциального распределения Raspred1 со значением среднего 0.8 и равномерного распределения Raspred2 в интервале (0.0, 0.2) получена следующая диаграмма.

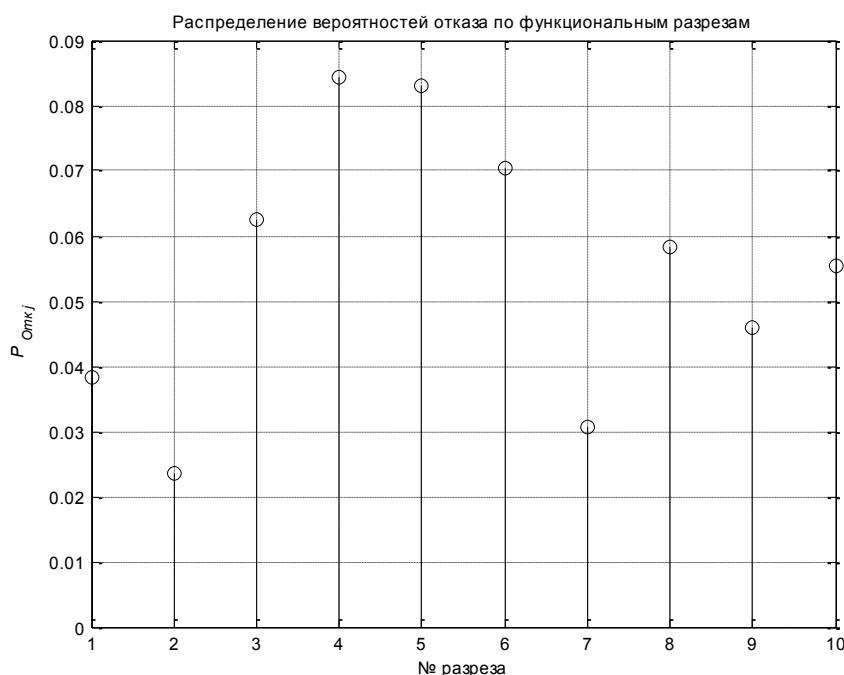


Рисунок 2 – Распределение оценок вероятностей отказов

Из зависимостей следует, что все функциональные модули содержат ошибки и нуждаются в доработке, наихудшим образом дело обстоит для модулей №№ 4 – 6. Если изменения в ПО не вносить, то вероятность безотказной работы, рассчитанная, согласно (2) будет убывать, как это видно на рисунке, приведённом ниже.

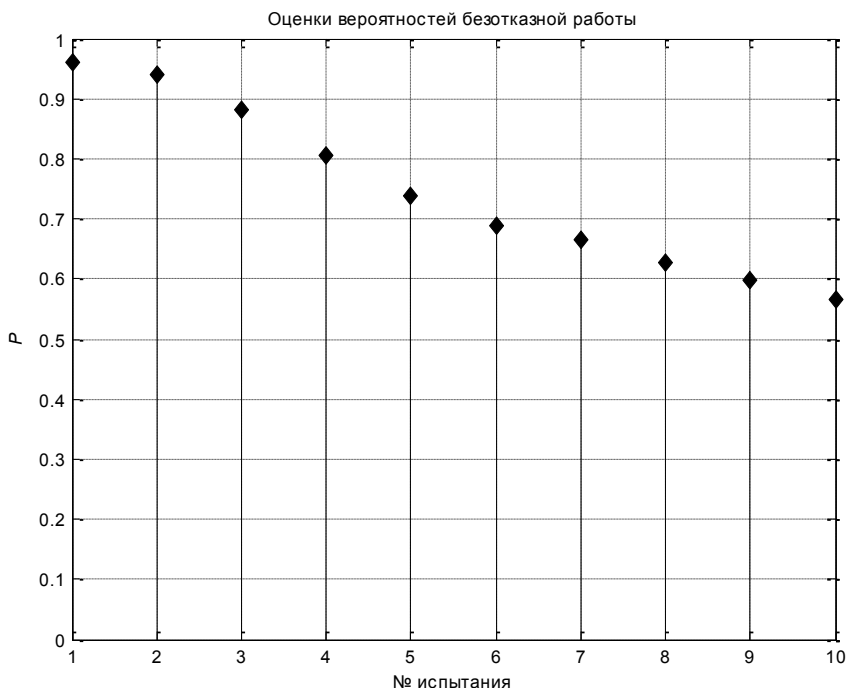


Рисунок 3 – Оценка надёжности в функции от номера прогона

Зависимость демонстрирует вполне очевидные вещи: если ошибки не устранять, то надёжность ПО будет низкой. Сводка по статистике моделирования и оценка Коркорэна при внесении исправлений по ходу тестирования ПО, рассчитанная по формуле (3), в предположении об устранении всех обнаруживаемых ошибок при модификации программного кода и отсутствии привносимых ошибок представлена на рисунке 3.

РЕЗУЛЬТАТЫ

Число функциональных разрезов 10
Число испытаний 1000
Оценка Коркорэна 0.99

Оценка надёжности Нельсона 0.594851

Рисунок 4 – Сводный отчёт по работе программы

3. ХОД РАБОТЫ

1. Получить вариант задания в виде типов распределений и их параметров. С учётом параметров, необходимых для генерации распределения соответствующего типа, построить схему моделирования, аналогичную представленной на рисунке 1. Некоторые процедуры генерации, используемые в

MatLab для решения задач в лабораторной работе, приведены в таблице 1.

Таблица 1 – Встроенные процедуры генерации псевдослучайных чисел

Тип распределения	Параметры вызова
Нормальное	<code>normrnd(<среднее>, <СКО>, 1, 1)</code>
Пуассоновское	<code>poissrnd(<среднее>, 1, 1)</code>
Равномерное	<code>unifrnd(<нижн_гр>, <верх_гр.>, 1, 1)</code>
Хи-квадрат	<code>chi2rnd(<среднее>, 1, 1)</code>
Экспоненциальное	<code>exprnd(<среднее>, 1, 1)</code>

2. На вкладках меню Simulations / Parameters задать значения полей: Stop Time установить равным 100.0, в поле Type установить значение Fixed-step, а в поле рядом discrete (no continuous states).

3. Написать программу для управления имитационной моделью.

Модель необходимо запускать в цикле, число повторений которого `nprusk` равно количеству функциональных разрезов.

Для решения этой задачи понадобятся следующие языковые средства и конструкции среды MatLab:

- `for k=1 : nprusk; ... end;` – конструкция цикла с заданным числом повторений;
- `sim('PO_imit');` – запуск модели с именем, заключённым в апострофы;
- `osybky.signals.values` – обеспечивает доступ к результатам работы имитационной модели, полученных посредством блока To Workspace.

4. Дополнить программу управления инструкциями для расчётов показателей надёжности (2) и (3)

Можно использовать:

- `mean(x)` – возвращает среднее значение массива чисел. Эту процедуру необходимо выполнять в цикле моделирования с каждым текущим вектором `osybky.signals.values`.
- `cumprod(x)` – вычисление накопленного произведения элементов вектора `x`, позволит изящно рассчитать последовательность R_n согласно (2);
- `sum(x)` – вычисление суммы вектора `x`.

При расчёте по формуле (3) необходимо учесть, что n определяется произведением Stop Time на число пусков имитационной модели `nprusk`. Параметр n_0 – есть разность между n и суммой элементов вектора усреднённых ошибок, полученных от имитационной модели. Также принимается гипотеза, что найденные ошибки гарантированно устраняются.

5. Дополнить программу управления операторами для построения графиков [6];

- `stem(x,y,<параметры линии>)` – позволит вывести график дискретных значений в виде набора вертикальных линий, например, операторы `stem(1:npush,p_otk,'k');` – использован для вывода рисунка 1,
`stem(1:npush,r_n,'MarkerFaceColor','black','Marker','diamond')` – для рисунка 2;
- `title('Текст');` – используется для вывода названия графика;
- `xlabel('Текст');` – для подписывания оси абцисс;
- `ylabel('Текст');` – для подписывания оси ординат;
- `grid;` – формирует сетку на экране;
- `hold` – фиксирует текущее изображение и позволяет пополнять его новыми элементами.

Управление шрифтами, цветом и с., смотри справочную и иную литературу среде MatLab.

6. Построить окно итогового отчёта.

Указанная процедура описана в [6] и включает ряд шагов.

- `axis('off');` – уборка из графического окна подокна с осями координат;
- `h1=text(<гор>,<вер>,'Текст','FontSize',20);` – выводение текста в позицию (<гор>,<вер>) окна кеглем № 20;
- `h1=text(<гор>,<вер>,sprintf('R=%g',r_c),'FontSize',12);` – выводение текста и цифровых значений, формат вывода аналогичен форматам языка Си.

7. Оформить отчёт.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое “прогон”?
2. Как вы понимаете термин “рабочий отказ”?
3. Как вы объясните, что означает термин “функциональный разрез”?
4. Почему для моделирования отказа ПО нам нужны два генератора псевдослучайных чисел?
5. Поясните назначение отдельных блоков схемы моделирования
6. Каким образом осуществляется связь имитационной модели и обрабатывающей программы в MatLab?
7. В чём сходство и различие в оценке надёжности ПО для моделей Нельсона и Коркорэна?
8. Какие еще модели надёжности ПО Вам известны [4]?
9. Какой вид, по Вашему мнению, может иметь распределение вероятностей ошибок по функциональному разрезу?

5. ВАРИАНТЫ ЗАДАНИЙ

Вариант задания выбирается как остаток от деления двух последних цифр зачётной книжки по модулю 25. Параметры распределения согласовываются с преподавателем.

№	Тип распределения 1	Тип распределения 2
0	Нормальное	Нормальное
1	Нормальное	Пуассоновское
2	Нормальное	Равномерное
3	Нормальное	Хи-квадрат
4	Нормальное	Экспоненциальное
5	Пуассоновское	Нормальное
6	Пуассоновское	Пуассоновское
7	Пуассоновское	Равномерное
8	Пуассоновское	Хи-квадрат
9	Пуассоновское	Экспоненциальное
10	Равномерное	Нормальное
11	Равномерное	Пуассоновское
12	Равномерное	Равномерное
13	Равномерное	Хи-квадрат
14	Равномерное	Экспоненциальное
15	Хи-квадрат	Нормальное
16	Хи-квадрат	Пуассоновское
17	Хи-квадрат	Равномерное
18	Хи-квадрат	Хи-квадрат
19	Хи-квадрат	Экспоненциальное
20	Экспоненциальное	Нормальное
21	Экспоненциальное	Пуассоновское
22	Экспоненциальное	Равномерное
23	Экспоненциальное	Хи-квадрат
24	Экспоненциальное	Экспоненциальное

6. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Боэм Б.В. Характеристики качества программного обеспечения [Текст]: / Б.В. Боэм. – М.: Мир, 1981.
2. Nelson E.C. Statistical Basic for Software Reliability Assessment, TRW-SS-73-03, March 1973.
3. Corcoran W.J. Estimating Reliability after Corective Action / W.J. Corcoran, H. Weingarten, P.W. Zehna. Management Science, 10, No 4, p.787 – 795 (July 1954).
4. Тейер Т. Надёжность программного обеспечения. / Т. Тейер, М. Липов, Э. Нельсон. – М.: Мир, 1981.

5. Гультяев А.К. : MatLab 5.2: имитационное моделирование в среде Windows [Текст]: / А.К. Гультяев СПб.: КОРОНА принт, 1999. – 288 с.
6. Лазарев Ю.Ф. MatLab 5.x. [Текст]: / Ю.Ф. Лазарев. – К.: Издательская группа ВНУ, 2000. – 384 с.

ЛАБОРАТОРНАЯ РАБОТА № 2

ИССЛЕДОВАНИЕ ТЕХНИЧЕСКИХ АСПЕКТОВ ПРОЕКТИРОВАНИЯ В СИСТЕМНОЙ ИНЖЕНЕРИИ

1. Цель работы

- Получить практические навыки формирования комплекса технических средств.
- Получить практические навыки в проведении количественного анализа вычислительных систем.

2. Теоретические сведения

При решении задачи структурного синтеза в системной инженерии целесообразно уметь выбирать и оценивать средства ввода и регистрации, обработки, передачи данных, т.е. решать проблему обеспечения процессов сбора, обработки, передачи данных в ИС.

2.1. Расчётные выражения для выбора средств ввода и регистрации

При выборе средств ввода и регистрации данных используются следующие оценки.

Общее количество средств регистрации:

$$N_{\text{общ}} = \sum_{i=1}^{n_p} N_i \cdot K_H, \quad (1)$$

где n_p – количество пунктов регистрации данных; N_i – количество средств, установленных в i -м пункте; K_H – коэффициент, который учитывает неравномерность поступления данных. Этот коэффициент определяется так.

$$K_H = \frac{t_{CM}}{t_{CM} - t_O}, \quad (2)$$

где t_{CM} – длительность смены (нахождения терминала во включённом состоянии в течении какого-либо фиксированного кванта времени), t_O – среднее время отсутствия данных в течение времени t_{CM} .

Для определения количества средств N_i , используется выражение

$$N_i = \sum_{j=1}^e \frac{M_{Pj}(1 + K_{KH})}{B_{i,j} \cdot t_{\text{доп}j}}, \quad (3)$$

где e – число видов регистрируемых данных; M_{Pj} – объём массива регистрируемой информации; K_{KH} – $B_{i,j}$ – нормативная характеристика производительности i -м устройством j -ых типов данных; $t_{\text{доп}j}$ – допустимое время приёма j -ого типа данных.

Выбранный состав средств должен удовлетворять условию

технологического цикла регистрации данных вида

$$Ц_p \leq Ц_{фсу} - Ц_{об} - T_{пи}, \quad (4)$$

где $Ц_{фсу}$ – цикл функционирования системы управления; $Ц_{об}$ – цикл обработки данных; $T_{пи}$ – время передачи информации.

Цикл работ по регистрации данных $Ц_p$, в свою очередь, есть сумма времён, определяемых функцией от продолжительности работы алгоритмов, составляющих технологический цикл.

$$Ц_p = \sum f(F_j, R_j, A_j, D_j), \quad (5)$$

где n – число выполняемых алгоритмов; F_j – время работы алгоритма формирования (фиксации) данных; R_j – время работы алгоритма регистрации (внесения) данных; A_j – время работы алгоритма накопления (аккумулирования) данных; D_j – время работы алгоритма задержки времени перед передачей на сервер. Указанные алгоритмы могут выполняться как программно, так и вручную.

При ручном способе формирования и внесения первичных данных

$$Ц_p = \sum_{j=1}^e \left(\frac{F_j}{N_F} + \frac{R_j}{N_R} + \frac{A_j}{N_A} + D \right), \quad (6)$$

где N_F – количество источников данных; N_R – количество пунктов регистрации (удалённых терминалов); N_A – количество пунктов накопления информации (локальных станций). В выражении (6) первые два слагаемых определяют производительность труда персонала или устройств, а вторые два – цикл функционирования системы в целом.

При комбинированном способе формирования данных получаем следующее

$$Ц_p = \sum_{j=1}^e \left(\frac{F_j}{N_F} + \frac{R_j + A_j}{N_{R,A}} + D \right). \quad (7)$$

Если данные формируются в полностью автоматическом режиме, то имеем

$$Ц_p = \sum_{j=1}^e \left(\frac{F_j + R_j}{N_{F,A}} + \frac{A_j}{N_A} + D \right). \quad (8)$$

2.2. Расчётные выражения для выбора средств передачи

Основное условие, которое можно соблюдать при выборе, чтобы фактическое время, требуемое для передачи данных $t_{п}$ было меньше некоторого заданного допустимого времени $t_{доп}$.

Необходимое количество каналов связи определяется по формуле

$$N_{кан} = \sum_{i=1}^d \frac{n_i}{r_i} + N_{рез}, \quad (9)$$

где d – количество датчиков; n_i – количество датчиков i -го вида; r_i – количество датчиков i -го вида присоединяемых к одному каналу; $N_{рез}$ – количество резервных каналов, на перспективу развития системы.

Суммарная разрядность определяется выражением

$$W_{\Pi} = \sum_{i=1}^{N_{КАН}} r_i \cdot W_i + W_{PEЗ}, \quad (10)$$

где r_i – количество датчиков, присоединённых к i -му каналу; W_i – разрядность датчиков, присоединённых к i -му каналу; $W_{PEЗ}$ – число резервных разрядов.

Объём передаваемой информации M_{Π} есть произведение суммарной разрядности на частоту передачи

$$M_{\Pi} = W_{\Pi} \times f_{\Pi}. \quad (11)$$

Время передачи информации для различных типов аппаратуры зависит от объёма передаваемых данных и составляет

$$t_{\Pi} = \frac{M_{\Pi}}{V_{\Pi i} \cdot R_v} + t_c, \quad (12)$$

где $V_{\Pi i}$ – скорость передачи по i -му типу аппаратуры; R_v – коэффициент снижения скорости передачи для повышения точности передаваемых данных ($0,5 < R_v < 0,7$); t_c – время, затрачиваемое на установление соединения.

Количество линий передачи может быть оценено как

$$N_{\text{л}} \geq \frac{t_{\Pi}}{t_{\text{доп}}}. \quad (13)$$

2.3. Расчётные выражения для выбора средств обработки

Необходимо сравнить степень соответствия параметров P_C – сравниваемой вычислительной системы и P_T – требуемых, полученных в результате расчётов. Соответствия могут быть заданы совокупностью баллов $\{b_i\}$, полученных в результате экспертного оценивания и относительного соизмерения сравниваемых величин i -х показателей

$$b_i = \frac{P_{C,i}}{P_{T,i}} \cdot z_i + \left(1 - \frac{P_{C,i}}{P_{T,i}}\right) \cdot z_{д,i}, \quad (14)$$

где z_i – коэффициент значимости i -го показателя; $z_{д,i}$ – коэффициент допустимости несоответствия i -го параметра по отношению к требуемому значению.

Оценка удовлетворения всей системы средств обработки производится по одной из формул

$$\eta_{\text{уд}} = \begin{cases} \sum_{i=1}^n b_i, & \text{а)} \\ \frac{1}{n} \sum_{i=1}^n b_i, & \text{б)} \\ \prod_{i=1}^n b_i, & \text{в)}. \end{cases} \quad (15)$$

Выражения 15, а и 15, б имеют тенденцию к сглаживанию резких отклонений расчётных параметров от желаемых. При использовании

выражения 15, в, эти отклонения оказывают существенное воздействие. Окончательный выбор производится по максимальному значению $\eta_{уд}$. Затем по отобранному варианту оценивается максимальная загрузка обрабатывающего комплекса

$$V = \frac{M_o}{T_p \cdot K_{доп}}, \quad (16)$$

где M_o – объём массива обрабатываемой информации; T_p – время решения задач; $K_{доп}$ – коэффициент дополнительного времени.

3. ХОД РАБОТЫ

1. Получить вариант задания. Провести расчеты по приведенным формулам.
2. Написать программу для расчёта.
4. Дополнить программу конструкциями для вывода графиков.
5. Оформить отчёт, защитить результаты выполнения лабораторной работы.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое цикл работ по регистрации данных?
2. Какое ограничение необходимо удовлетворить при выборе средств передачи?
3. Как оценить количество каналов связи?
4. Как рассчитать количество линий передачи?
5. От чего зависит, насколько система удовлетворяет требуемым параметрам?

5. ВАРИАНТЫ ЗАДАНИЙ

Вариант задания выбирается студентом индивидуально, в соответствии с тематикой выпускной (квалификационной) магистерской работы по согласованию с её руководителем.

ЛАБОРАТОРНАЯ РАБОТА № 3 ИССЛЕДОВАНИЕ АСПЕКТОВ КОЛИЧЕСТВЕННОГО АНАЛИЗА КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ В СИСТЕМНОЙ ИНЖЕНЕРИИ

1. Цель работы

- Получить практические навыки формирования количественных оценок

комплекса технических средств.

- Получить практические навыки в проведении количественного анализа вычислительных систем.

2. Теоретические сведения

В основе анализа комплекса технических средств лежит определение соответствия объёмов работ и временных ограничений по количеству и производительности технических средств.

Основным ограничивающим временным фактором является цикл работ (Ц) обработки информации. Под этим термином понимается промежуток времени от начала поступления данных на обработку T_H до срока готовности задачи T_G :

$$Ц = T_G - T_H. \quad (1)$$

В свою очередь, цикл работ будет складываться из допустимых продолжительностей выполнения подготовительных операций T_{II} и времени решения задач на ПЭВМ (вычислений) T_B . Продолжительность решения задач на ПЭВМ зависит от реализации основных этапов и режимов обработки информации.

Композиционно время решения задачи решения разбивается на этапы

$$t = t_I + t_P + t_E + t_O, \quad (2)$$

где t_I – ввод (загрузка, input) программы и данных; t_P – процессорное время (processing), затрачиваемое на счёт непосредственно; t_E – время, затрачиваемое на обмен в процессе счёта с внешними (external) устройствами; t_O – формирование и вывод (output) результатов работы.

Время ввода определяется по элементарной формуле

$$t_I = \frac{M_I}{v_I}, \quad (3)$$

где M_I – объём массива вводимых данных; v_I – быстродействие устройства.

Процессорное время может рассчитываться двояко, в зависимости от степени известности информации об организации вычислительного процесса. В первом случае

$$t_P = \bar{t}_{op} \cdot k_{CT} \cdot M_I, \quad (4)$$

где $\bar{t}_{op} = \sum_j t_j \cdot q_j$ – среднее время выполнения операции, k_{CT} – коэффициент соотношения операций, показывающий, какое количество операций приходится на байт введённой информации; t_j – время выполнения j -й операции; q_j – удельный вес j -й операции.

Для определения коэффициентов соотношения операций и удельных весов операций задачи разбиваются на типовые группы.

I. Задачи перегруппировки и упорядочивания информации, нахождения сумм, выборки, сечений по относительно простым критериям. Удельный вес

логических операций достигает 70%, арифметических – до 30%. Коэффициент соотношения находится в пределах 50 – 100 операций на байт введенной информации.

II. Манипуляция несколькими массивами, часть которых находится на внешних запоминающих устройствах. Объем арифметических операций незначителен: нахождение сумм, парных произведений, операции с матрицами. Удельный вес логических операций достигает 90 – 95%, арифметических – до 5 – 10%, $k_{CT} \in (1000 - 2000)$ операций на байт.

III. Задачи экстремального характера: численная реализация сложных математических методов, изобилует циклами, итерациями и т. п. Удельный вес арифметических операций достигает 90%, а удельный вес логических – всего около 10%, $k_{CT} \in (1000 - 3000)$ операций на байт.

Во втором случае, если априори известен алгоритм решения задачи, то расчёты целесообразно вести по формуле

$$t_p = \sum_j t_j \cdot h_j, \quad (5)$$

где h_j – количество j -х операций применённых в реализации алгоритма.

Затраты времени на обмен с внешними устройствами рекомендуется определять по формуле

$$t_E = \frac{k_{rw}}{v_{rw}} \sum_j f_j \cdot V_j, \quad (7)$$

где k_{rw} – коэффициент, учитывающий контрольные операции, при проведении операций считывания или записи (например, для контрольного суммирования $k_{rw} = 2$); v_{rw} – скорость чтения или записи; f_j – количество обращений к внешним устройствам; V_j – объём считываемой или записываемой порции данных. Коэффициенты, находящиеся в (7) перед знаком суммы, могут быть, в случае разнообразия устройств ввода и вывода, использованы в качестве множителей под знаком.

Время вывода результата определяться быстродействием конечного устройства как отношение

$$t_o = \frac{M_o}{v_o} = \frac{M_I \cdot K_{IO}}{v_o}, \quad (8)$$

где M_o – объём массива выводимых данных; v_o – быстродействие устройств; M_I – объём массива вводимых данных; K_{IO} – коэффициент соотношения выводимой информации по отношению к объёму входной.

Общее время решения последовательности задач составит

$$t_\Sigma = \sum_i t_i, \quad (9)$$

где t_i определяется (2) для каждой задачи.

Отсюда потребность в вычислительных устройствах составляет

$$N \geq \frac{Ц}{t_\Sigma}. \quad (10)$$

3. ХОД РАБОТЫ

1. Получить вариант задания. Провести расчеты по приведенным формулам.
2. Написать программу для расчёта.
4. Дополнить программу конструкциями для вывода графиков.
5. Оформить отчёт, защитить результаты выполнения лабораторной работы

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое цикл работ обработки информации?
2. Для чего задачи, решаемые на ЭВМ, разбиваются на типовые группы
3. Какова роль соотношений между видами операций при расчёте процессорного времени?
4. Что такое “время решения последовательности задач”?

5. ВАРИАНТЫ ЗАДАНИЙ

Вариант задания выбирается студентом индивидуально, в соответствии с тематикой выпускной (квалификационной) магистерской работы по согласованию с её руководителем.

ЛАБОРАТОРНАЯ РАБОТА № 4

ИССЛЕДОВАНИЕ ЭКОНОМИЧЕСКИХ АСПЕКТОВ АНАЛИЗА КОМПЛЕКСА ТЕХНИЧЕСКИХ СРЕДСТВ В СИСТЕМНОЙ ИНЖЕНЕРИИ

1. Цель работы

- Получить практические навыки определения экономических показателей комплекса технических средств.
- Получить практические навыки в проведении стоимостного анализа вычислительных систем.

2. Теоретические сведения

Цель стоимостного анализа заключается в обеспечении наибольшей эффективности на единицу затрат машинной обработки информации при соблюдении заданных условий.

Одним из методов проведения экономического анализа комплекса технических средств является оценка себестоимости информации C . Практически, себестоимость складывается из ряда статей, из которых основными являются: прямые расходы по заработной плате; административно-управленческие и общехозяйственные расходы; общепроизводственные расходы.

Расходы по заработной плате есть вся заработная плата производственного с отчислениями в различные фонды, например, на социальное страхование.

Административно-управленческие расходы включают зарплату административно-управленческого персонала и обслуживающего персонала, расходы на командировки, канцелярские и прочие управленческие расходы.

Общепроизводственные расходы условно делятся на три группы: расходы на эксплуатацию оборудования (электроэнергия, текущий ремонт оборудования и т. п.); прочие расходы (стоимость расходных материалов, бумаги, тонера, расходы на подготовку и повышения квалификации кадров); распределяемые расходы (охрана труда, содержание зданий, износ малоценного инструментария и инвентаря, транспортные расходы).

Суммарный характер расходов не позволяет определить расходы по отдельной решаемой задаче. Поэтому объем расходов на практике распределяют пропорционально использованию машинного времени или объема перерабатываемой информации.

В первом случае, это себестоимость машинного времени полезной работы вычислительной системы, то есть

$$S_{MT} = \frac{C}{T_p}, \quad (1)$$

где C – себестоимость, полученная суммированием ряда статей, упомянутых выше, T_p – полезное время работы.

Второй подход в качестве калькуляционной единицы использует единицу вводимой информации. Себестоимость обработки единицы вводимой информации будет равна

$$S_m = \frac{C}{M}, \quad (2)$$

где M – объём перерабатываемой информации.

По предельным объёмом перерабатываемой информации M_{Π} будем понимать такой объём, который может быть обработан за некоторый отрезок (период) времени τ_{BM} функционирования системы. Указанный объём может быть рассчитан по формуле

$$M_{\Pi} = \frac{\tau_{BM} \cdot F_{ДН} \cdot K_{И}}{\bar{t} \cdot K_{ДОП}}, \quad (3)$$

где $F_{ДН}$ – дневной (суточный) фонд работы системы, обычно задаётся в часах; $K_{И}$ – коэффициент использования вычислительной системы; \bar{t} – среднее время обработки единицы информации; $K_{ДОП}$ – коэффициент дополнительного времени.

Среднее время обработки единицы информации рассчитывается как сумма

$$\bar{t} = \sum_{N_I} t_I \cdot q_I + \sum_{N_P} t_P \cdot K_P \cdot q_P + \sum_{N_O} t_O \cdot K_O \cdot q_O. \quad (4)$$

где N_I – количество разновидностей способов ввода информации; t_I – время ввода единицы информации; q_I – удельный вес разновидности ввода; N_P – количество групп сложности задач; t_P – среднее время выполнения операции; q_P – удельный вес задач соответствующей сложности; K_P – коэффициент соотношения операций; N_O – количество разновидностей способов вывода информации; t_O – время вывода единицы информации; q_O – удельный вес вида выводимой информации; K_O – коэффициент соотношения вывода.

На рисунке 1 показан принцип анализа систем для комплексов 1 и 2. Комплекс 1 имеет меньшие эксплуатационные расходы, чем комплекс 2, но он, по сравнению с ним, является маломощным, поскольку имеет меньший объём перерабатываемой информации. Желание увеличить этот объём приведёт к увеличению мощностей вычислительного оборудования, что автоматически приведёт к росту эксплуатационных расходов, и к увеличению стоимости обработки.

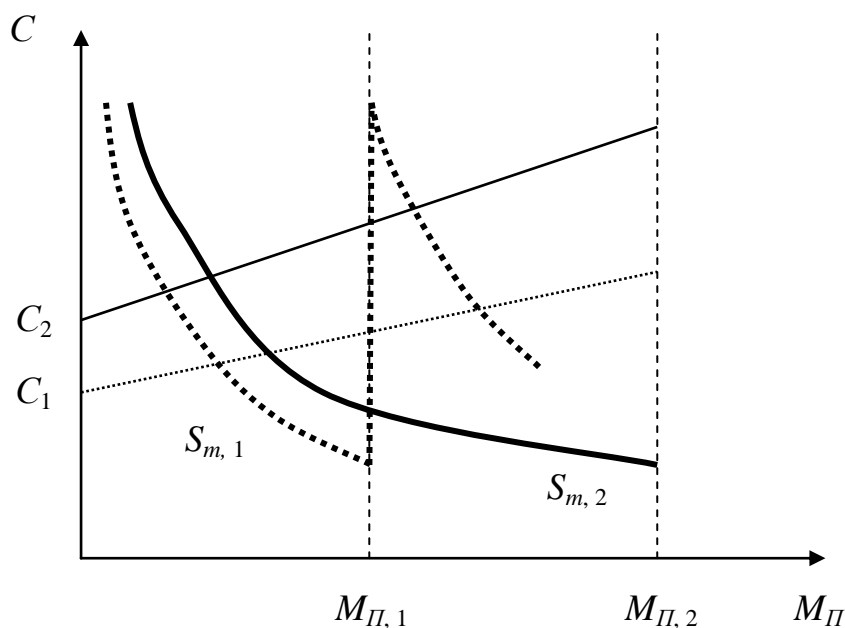


Рисунок 1 – Качественные графики анализа комплексов

Скачок на графике для линии $S_{m,1}$ объясняется как раз этой причиной. Таким образом, слева от $M_{П,1}$ будет эффективен 1-й комплекс, а от $M_{П,1}$ до $M_{П,2}$ второй. После границы $M_{П,1}$ аналогичный скачок по тем же причинам будет претерпевать $S_{m,2}$.

3. ХОД РАБОТЫ

1. Получить вариант задания. Провести расчеты по приведенным формулам.
2. Написать программу для расчёта.
4. Дополнить программу конструкциями для вывода графиков.
5. Оформить отчёт, защитить результаты выполнения лабораторной работы

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каковы цели и задачи стоимостного анализа?
2. Что такое административно-управленческие расходы?
3. Что такое общепроизводственные расходы?
4. Почему для изначально маломощного комплекса при возрастании объёма обрабатываемой информации себестоимость изменяется скачкообразно?

5. ВАРИАНТЫ ЗАДАНИЙ

Вариант задания выбирается студентом индивидуально, в соответствии с тематикой выпускной (квалификационной) магистерской работы по согласованию с её руководителем.

ЛАБОРАТОРНАЯ РАБОТА №5

ФОРМУЛИРОВКА И АНАЛИЗ ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ

1. Цель работы

- ознакомиться с процессом формирования и анализа требований;
- осуществить сбор, формулировку и документирование требований к рассматриваемой системе.

Время: 4 часа

2. Краткие теоретические сведения

Системная инженерия – это практическое применение научных, инженерных и управленческих навыков, необходимых для преобразования операционных требований в описание конфигурации системы, которая наилучшим образом удовлетворяет этим требованиям. Это общий процесс решения проблем, который применяется ко всему техническому управлению в проекте, посвященном разработке системы, предоставляя механизм формулирования и совершенствования определений изделий и процессов системы.

Системная инженерия включает в себя пять функций.

- определение проблемы – указание потребностей и ограничений путем анализа требований и взаимодействия с заказчиком;
- анализ решений – выделение набора возможных способов удовлетворения потребностей и ограничений, их анализ и выбор оптимального;
- планирование процессов – определение задач, которые должны быть выполнены, объема ресурсов и затрат, необходимых для создания изделия, очередности задач и потенциальных рисков;
- контроль процессов – определение методов мониторинга проекта и процессов, измерение прогресса, оценка промежуточных изделий и принятие по мере необходимости корректирующих действий;
- оценка изделий – определение качества и количества создаваемых изделий путем оценочного планирования, тестирования, демонстрации, анализа, верификации и контроля. Системная инженерия определяет систему как совокупность компонентов (физические системы, аппаратное обеспечение, программное обеспечение и персонал), которые совместно реализуют предполагаемые функции конечного продукта. Разработка продуктов – это логическая последовательность оценок затрат на циклы жизни продукта, управление исследованием технических компромиссных решений, анализ эффективности затрат, моделирование эффективности, проектирование архитектуры и процесса изготовления. Все вместе эти шаги преобразуют цели в параметры эффективности и предпочтительную конфигурацию системы.

На рис.5.1 показан типичный итеративный процесс системной инженерии. В его рамках выполняются попытки оценить варианты системной архитектуры и выбрать из них наилучшие.

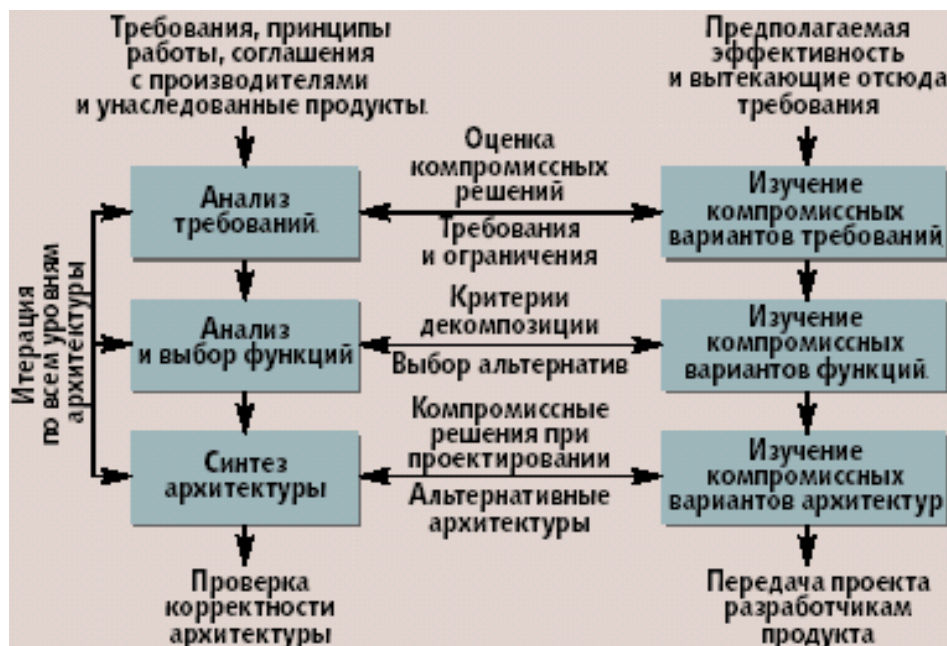


Рисунок 5.1 Типичный процесс системной инженерии при определении архитектуры

Первый шаг в любой активности – определение требований, их систематизация, документирование в виде спецификации системных требований. (перечисленная совокупность действий, дополненная анализом, выявлением противоречий, неполноты, разрешением конфликтов в процессе разработки – в англоязычной литературе называют инженерией требований).

Процесс формирования и анализа требований проходит через ряд этапов:

- разработка требований:
 - выявление требований;
 - анализ требований;
 - спецификация требований;
 - проверка требований;
- управление требованиями.

Выявление и анализ требований заключается в выявлении заинтересованных лиц (в англоязычной литературе – стейкхолдеров – физические лица или организации, имеющие права, долю, требования или интересы относительно системы или её свойств, удовлетворяющих их потребностям и ожиданиям (ISO/IEC 15288:2008, ISO/IEC 29148:2011)). Опрос стейкхолдеров является широко используемой техникой при сборе требований.

Собранные требования могут быть отнесены к следующим типам:

– **требования клиентов**, то есть тех, кто выполняет основные функции системного проектирования, со специальным акцентом на пользователе системы как ключевом клиенте. Пользовательские требования определяют

главную цель системы и, как минимум, ответят на следующие вопросы: где система будет использоваться? (требования эксплуатации или развёртывания); как система достигнет целей миссии? (профиль миссии или сценарий); какие параметры системы являются критическими для достижения миссии? (требования производительности); как различные компоненты системы должны использоваться? (сценарии использования); насколько эффективной должна быть система для выполнения миссии? (требования эффективности); как долго система будет использоваться? (эксплуатационный жизненный цикл); каким окружением система должна будет эффективно управлять? (окружающая среда);

- **функциональные требования** объясняют, что должно быть сделано. Они идентифицируют задачи или действия, которые должны быть выполнены, определяют связь входа/выхода в поведении системы;

- **нефункциональные требования** определяют критерии работы системы в целом, а не отдельные сценарии поведения (системные свойства такие как производительность, удобство сопровождения, расширяемость, надёжность, средовые факторы эксплуатации);

- **производные требования** – подразумеваются или преобразованы из высокоуровневого требования.

Методы документирования требований (составления спецификация требований):

- **списки требований** – традиционный способ документировать требования (используются и по сей день, хотя крайне неэффективны в современном анализе, в сложной системе такие списки требований могут занимать сотни страниц);

- **пользовательские истории** – определяют требования обычным языком;

- **измеримые цели** – рассматривается составленный список требований как подсказки и постоянно задается вопрос: «почему?», до тех пор, пока не будут выявлены истинные цели; разрабатываются тесты, измеряющие, какой уровень каждой цели был достигнут. Как только набор критических, измеримых целей установлен, быстрое прототипирование и короткие этапы разработки могут дать реальную ценность ещё до окончания проекта (являются практикой целеориентированной системной инженерии);

- **прототипирование** – построение макета (модели) системы. Макеты дают возможность представить систему, которая ещё не построена;

- **сценарии использования** – варианты использования (англ. Use Case) – техника для документации потенциальных требований для создания новой системы или изменения существующей. Каждый вариант описывает один или несколько способов взаимодействия системы с конечным пользователем или другой системой, для достижения определённой цели. Варианты использования обычно избегают технического жаргона, предпочитая вместо этого язык конечного пользователя или эксперта в данной области (являются практикой объектно-ориентированной системной инженерии и моделиориентированной системной инженерии);

Чтобы определить и выразить различные аспекты процесса разработки, в том числе и этапа формирования и анализа требований, применяются такие методики, как:

- сети Петри (помогают определить параллельность и синхронизацию);
- машины с конечным числом состояний (состояния и режимы);
- структурный анализ (потoki данных);
- PSL/PSA (Problem Statement Language/Problem Statement Analysis);
- язык Unified Modeling Language (OMG Unified Modeling Language Specification, Object Management Group, Sept. 2001).

Задание на лабораторную работу №5

1. Определить требования пользователя к рассматриваемой системе.
2. Определить требования к рассматриваемой системе в рамках целеориентированной системной инженерии.
3. Определить обязанности рассматриваемой системы, то есть указать, что должна делать система (функциональные требования) (модель «черный ящик» – зафиксировать и перечислить входные и выходные параметры рассматриваемой системы, связи со средой функционирования), определить связь входа/выхода в поведении системы.
4. Определить нефункциональные требования (критерии работы системы).
5. Построить сценарии использования рассматриваемой системы.

Содержание отчета по лабораторной работе №5

1. Требования пользователя к рассматриваемой системе (список требований или пользовательские истории).
2. Список измеримых целей системы.
3. Функциональные требования к рассматриваемой системе (входные/выходные параметры, их связь, внешние воздействия).
4. Нефункциональные требования (критерии, которым должна соответствовать функционирующая система).
5. Сценарии поведения системы.

Контрольные вопросы

1. Какова основная задача системной инженерии?
2. Перечислить этапы (фазы) процесс формирования и анализа требований?
3. Какова цель процесса определения требований в соответствии с ГОСТ Р ИСО/МЭК 15288-2005?
4. Перечислить и раскрыть суть методов документирования требований.

ЛАБОРАТОРНАЯ РАБОТА № 6

АНАЛИЗ ПОДХОДОВ К ФОРМАЛИЗАЦИИ ОПИСАНИЯ СЛОЖНЫХ СИСТЕМ

1. Цель работы

- Получить практические навыки формализации задач системной инженерии.
- Получить практические навыки в проведении анализа и формализации требований к ИС.

2. Теоретические сведения

Важную роль при анализе и реализации требований к системе играет их специфицирование, для создания систем программного уровня – Software Requirements Specification (SRS). Следует отметить, что часто под термином "система" подразумевается программная система или приложение в разработке. Это может быть крупная коллекция множества компонентов программного и аппаратного обеспечения, одно приложение или компонент программного обеспечения внутри более крупной системы. Во всех этих случаях модель требований описывает поведение, видимое вне системы (через пользовательский интерфейс или API) [1, 2, 3].

Модель описания требований уровня UML во многих случаях не отражает особенности построения узкоспециализированных классов систем в широкой трактовке. Исходя из методики, предложенной А.В. Дабагяном [4, 5], требования характеризуются физическими параметрами, а требования, отличающиеся хотя бы одним признаком или одним значением его параметра, должны быть отнесены к различным классам.

Поле заявок представлено в многомерном евклидовом пространстве. Размерность евклидова пространства равна m , что соответствует отдельным признакам заявки. В поле вектора Z некоторую заявку можно представить в виде вектора Z_i с координатами Z_{ij} , $j \in \{1, \dots, m^1\}$. Полем требований в m^1 -мерном пространстве вектора Z называется замкнутая $m^1 - 1$ -мерная область, границы которой ортогональны осям евклидова пространства, где заданы векторы заявок Z . В той же работе введены определения минимальной и максимальной j -ых границ поля требований.

$$T_i^{\chi} = [\tau_{1j}^{\chi}, \dots, \tau_{ij}^{\chi}, \dots, \tau_{i(m^1-1)}^{\chi}]^T, \quad Z^{\chi} = [z_1^{\chi}, \dots, z_j^{\chi}, \dots, z_{m^1}^{\chi}]^T, \quad (1)$$

где i – номер требования; j – номер параметра требования; $(m^1 - 1)$ – размерность вектора требований; T – знак транспонирования; Z^{χ} – заявка класса χ ; m^1 – число параметров.

$$Z^{\chi} = \bigcup_{\forall i} T_i^{\chi},$$

где i – номер требования класса χ в заявке Z^{χ} .

Ограничениями данных определений является то, что обслуживание требований одного класса предполагается по одинаковой технологии, а обслуживание требований, входящих в заявки различных классов – по различным технологиям. Однако, реализация *on-line* требований, связанная, в том числе с новыми заявками, не всегда может быть отнесена к существующим технологиям. Следовательно, целесообразной является предварительная оценка времени реализации требования, а также классификация требований и соотнесение их с возможными реализациями.

Фаза анализа и планирования требований в рамках технологии RAD - быстрой разработки приложений [6] связана со следующими работами: 1) определяются функции, которые должна выполнять разрабатываемая информационная система; 2) определяются наиболее приоритетные функции, требующие разработки в первую очередь; 3) проводится описание информационных потребностей; 4) ограничивается масштаб проекта; определяются временные рамки для каждой из последующих фаз; в заключение, 5) определяется сама возможность реализации данного проекта в установленных рамках финансирования, на имеющихся аппаратных и программных средствах.

При использовании методологии быстрой разработки приложений жизненный цикл информационной системы состоит из четырех фаз: • фаза анализа и планирования требований; • фаза проектирования; • фаза построения; • фаза внедрения.

Если реализация проекта принципиально возможна, то результатом фазы анализа и планирования требований будет список функций разрабатываемой информационной системы с указанием их приоритетов и предварительные функциональные и информационные модели системы.

Таким образом, анализ и формализация требований играет важную роль при анализе функционирования системы в целом.

Пример выполнения работы

Пусть требуется построить ИС по следующему описанию: конструкторское бюро по проектированию сложных технических объектов. Предполагается использование большого количества программных средств дизайнерского назначения и средств САПР¹. Предполагаются активная передача информации по локальной сети между конструкторами, имеется необходимость защиты особых данных. Особо важна скорость реализации процесса обработки изображений.

Задание: 1. Формализовать требования к системе на основе технологии RAD,
2. Формализовать описание системы в целом.

¹ Подготовленная в САПР полная схема является не просто набором чертежей, но и содержит информацию о соединениях всех элементов. С перечнем аппаратуры связаны данные о размещении, габаритах и схемах аппаратов. На их основе формируется база данных проекта, позволяющая использовать ее для создания других документов. Использование именно базы данных проекта, а не отдельных чертежей, позволяет значительно повысить эффективность выпуска рабочей документации проектными организациями.

Часть 1. Исходя из технологии RAD, проводится анализ требований:

1) определяются функции, которые должна выполнять разрабатываемая информационная система;

Из описания системы, определяется множество функций ИС. Глобальная функция: Φ – получение комплексного проекта в виде совокупности изображений.

Поддерживающие и обеспечивающие функции: $\Phi 1$ – обеспечение взаимодействия многих программ дизайнерского назначения и средств САПР; $\Phi 2$ – поддержка интенсивного обмена информацией по локальной сети между конструкторами; $\Phi 3$ – функция защиты особых данных;

$\Phi 4$ – поддержка высокой скорости реализации процесса обработки изображений.

$$\Phi: \Phi 1, \Phi 2, \Phi 3, \Phi 4 \quad (2)$$

2) определяются наиболее приоритетные функции, требующие разработки в первую очередь;

$\Phi 1$ – обеспечение взаимодействия многих программ дизайнерского назначения и средств САПР можно считать базовой функций разрабатываемой ИС, поскольку исходя из способа реализации $\Phi 1$ будут обеспечиваться остальные функции.

3) проводится описание информационных потребностей;

Информационными потребностями пользователя являются конкретные (фактические) значения требуемых параметров системы, которые задаются заказчиком проекта. Например, объемы текущих данных, требуемая скорость обмена данными, или время, необходимое на передачу данных.

Эти данные выбираются либо по согласованию с преподавателем, либо выбираются самостоятельно, исходя из среднестатистических данных, например, размера некоторого изображения САПР в байтах (формат чертежа A1), скорость обмена порядка 2Мбайт/сек и т.п.

4) ограничивается масштаб проекта;

Ограничения могут быть связаны с уточнениями описания модели. Для приведенного примера это могут быть ответы на вопросы: какой тип САПР используется? (От этого зависит размер картинки, стоимость разработки). Например:

1. САПР двумерного проектирования — «2D-3D Легкие — Нижний уровень»²

Эти САПР служат для выполнения почти всех работ с двумерными

² <http://www.automationlabs.ru/>

чертежами и имеют ограниченный набор функций по трехмерному моделированию. С помощью этих систем выполняются порядка 90% всех работ по проектированию. Область их работы — создание чертежей отдельных деталей и сборок. Платой за возросшие возможности является усложнение интерфейса и меньшее удобство в работе. Характерные представители таких САПР — AutoCAD, CADdy, CADMECH Desktop, MasterCAM, T-FlexCAD, OmniCAD, Компас-График.

2. САПР объемного моделирования «3D — Средний уровень»

По своим возможностям они полностью охватывают САПР «легкого веса», а также позволяют работать со сборками, по некоторым параметрам они уже не уступают тяжелым САПР, а в удобстве работы даже превосходят. Обязательным условием является наличие функции обмена данными (или интеграции). Это не просто программы, а программные комплексы, в частности, SolidWorks SolidEdge, Cimatron, Form-Z, Autodesk Inventor, CAD SolidMaster, и все еще продолжающий развиваться, Mechanical Desktop, DesignSpace.

3. САПР объемного моделирования «3D Тяжелые — Верхний уровень»

Эти системы применяются для решения наиболее трудоемких задач - моделирования поведения сложных механических систем в реальном масштабе времени, оптимизирующих расчетов с визуализацией результатов, расчетов температурных полей и теплообмена и т.д. Обычно в состав системы входят графические, модули для проведения расчетов и моделирования, постпроцессоры для станков с ЧПУ. Эти самые мощные САПР наиболее громоздки и сложны в работе, а также имеют значительную стоимость. Примерами «тяжелых» САПР могут служить такие продукты, как ADAMS, ANSYS, CATIA, EUCLID3, Pro/ENGINEER, UniGraphics.

Стоимость всех САПР соотносятся по уровням следующим образом:

- Нижний: \$500-\$2000 за рабочее место (AutoCAD, AutoCAD LT, Компас);
- Средний: \$2000-\$20000 (Inventor, Mechanical Desktop, SolidWorks);
- Верхний: более \$20000 (ProEngineer, Unigraphics).

Кроме того, ограничениями может быть уточнение требуемой скорости обмена данными и т.п.

5) определяется возможность реализации данного проекта в установленных рамках ограничений.

Этот этап служит для анализа требований на совместимость и непротиворечивость.

В примере: при ограничении ресурсов по финансированию проекта применение САПР верхнего уровня может привести к полной невозможности его реализации. Или ограничение на скорость обмена данными вследствие применения определенного типа аппаратуры может сказаться на реализации соответствующего требования.

Таким образом, анализируются функции и определяются количественные требования.

Формализация требований осуществляется посредством их символического представления согласно (1) и (2). **Функции системы:**

$$\Phi: \Phi_1, \Phi_2, \Phi_3, \Phi_4,$$

где Φ – получение комплексного проекта в виде совокупности изображений.

Φ_1 – обеспечение взаимодействия многих программ дизайнерского назначения и средств САПР; Φ_2 – поддержка интенсивного обмена информацией по локальной сети между конструкторами; Φ_3 – функция защиты особых данных; Φ_4 – поддержка высокой скорости реализации процесса обработки изображений.

Требования к системе:

Определим поле требований как вектор требований, их шкал и ограничений:

$$T_1^{\Phi_1} = [\tau_{11}^{\Phi_1}, \tau_{12}^{\Phi_1}],$$

где $T_1^{\Phi_1}$ – требования класса Φ_1 : $\tau_{11}^{\Phi_1}$ – наличие гетерогенной информационной среды; $\tau_{12}^{\Phi_1}$ – наличие интерфейса взаимодействия различных элементов гетерогенной информационной среды.

При этом, если $T_1^{\Phi_1} \in \overline{0,1}$, то $\tau_{11}^{\Phi_1}$, $\tau_{12}^{\Phi_1}$ могут быть трактованы как степени принадлежности к нечеткому множеству соответствия, а если $T_1^{\Phi_1} = \overline{0,N}$, где N – общее кол-во возможных значений $T_1^{\Phi_1}$, то $\tau_{11}^{\Phi_1}$, $\tau_{12}^{\Phi_1}$ имеют смысл цифровых количественных оценок; при $T_1^{\Phi_1} \in 0,1$, то $\tau_{11}^{\Phi_1}$, $\tau_{12}^{\Phi_1}$ – бинарные определители наличия или отсутствия признака требования: есть гетерогенная среда или нет.

$$T_1^{\Phi_2} = [\tau_{11}^{\Phi_2}, \tau_{12}^{\Phi_2}],$$

где $T_1^{\Phi_2}$ – требования класса Φ_2 : $\tau_{11}^{\Phi_2}$ – обеспечение заданной скорости обмена данными; $\tau_{12}^{\Phi_2}$ – обеспечение заданной надежности обмена данными. При ограничениях: $\tau_{11}^{\Phi_2} \geq 2 \text{ Мб/сек.}$, $0.8 \leq \tau_{12}^{\Phi_2} \leq 1$.

Аналогичным образом строятся множества

$T_1^{\Phi_3} \dots T_K^{\Phi_3} = [\dots], T_1^{\Phi_4} \dots T_L^{\Phi_4} = [\dots]$, где K, L – количество типов требований каждого класса Φ_i .

Часть 2. Формализация описания системы в целом.

В Таблице 1 приведены основные способы представления формального описания сложных систем.

Поскольку глобальная функция системы представляет собой формирование нового информационного объекта (Φ – получение комплексного проекта в виде совокупности изображений), можно использовать при описании сочетание п.4 и п.2 (Табл.1.).

Основные формы описания систем

№	Определение	Формальное описание	Описание
1	Система есть организованное множество	$S=(O, M)$, где O - оператор организации; M - множество	Упрощенное описание любой системы
2	Система есть множество элементов, свойств и отношений	$S=(\{m\}, \{n\}, \{r\})$, где t - элементы, n - свойства, r - отношения	Знаково-графовый подход
3	Система есть множество элементов, образующих структуру и обеспечивающих определенное поведение в условиях окружающей среды	$S=(\varepsilon, ST, BE, E)$, где ε - элементы, ST - структура, BE - поведение, E - среда	Структурно-поведенческий подход
4	Система есть множество входов, множество выходов, множество состояний, характеризуемых оператором переходов и оператором выходов	$S=(X, Y, Z, H, G)$, где X - входы, Y - выходы, Z - состояния, H - оператор переходов, G - оператор выходов	Учитываются все основные компоненты, рассматриваемые в автоматике.
5	Понятия модели F , связи SC , пересчета R , самообучения FL , самоорганизации FQ , проводимости связей CO и возбуждения моделей JN	$S=(F, SC, R, FL, FO, CO, JN)$	Определение применяется для описания нейрокибернетических систем.
6	Введен фактор времени и функциональных связей	$S=(T, X, Y, Z, \Omega, V, \eta, \varphi)$, где T - время, X - входы, Y - выходы, Z - состояния, Ω - класс операторов на выходе, V - значения операторов на выходе, η - функциональная связь в уравнении $y(t_2) = \eta(x(t_1), z(t_1), t_2)$, φ - функциональная связь в уравнении $z(t_2) = \varphi(x(t_1), z(t_1), t_2)$.	Определение системы, которым оперируют в теории автоматического управления
7	Организационные системы	$S9=(PL, RO, RJ, EX, PR, DT, SV, RD, EF)$, где PL - цели и планы, RO - внешние ресурсы, RJ - внутренние ресурсы, EX - исполнители, PR - процесс, DT - помехи, SV - контроль, RD - управление, EF - эффект.	Организационное управление

Таким образом, возможно применить следующее обобщенное описание

$$S=(X, Y, Z, H, G, \{m\}, \{n\}, \{r\}),$$

где X - входы, Y - выходы, Z - состояния, H - оператор переходов, G - оператор выходов, t - элементы, n - свойства, r - отношения между элементами.

Пример. Гидрометеорологическая система с возможностью сбора, анализа и хранения данных. Опишем функционирования этой ИС типа с требованиями.

Проектируемая система производит преобразование

$$Y_t = F(X_t, T_t),$$

где X_t – текущее состояние входного объекта системы в момент времени t ; Y_t – текущее состояние выходного объекта системы в момент времени t ; T_t – текущее требование к системе;

F – функция преобразования входного объекта системы.

Система может быть описана как

$$S = \left[\bigcup_{n=1}^N \{X_i^{(n)} \mid (\delta_i \leq \delta^{X_i})\}, \{Y_j^{(k)} \mid (\theta_j \leq \theta^{Y_j})\}, \{T_t \mid T_i^{\Phi_j} = [\tau_{i1}^{\Phi_j}, \dots, \tau_{iI}^{\Phi_j}]\}, \Phi, (Z : z_i \geq z^*) \right],$$

где $\bigcup_{n=1}^N \{X_i^{(n)}\}$ – объединение множеств входных объектов (данных)

системы, $n = \overline{1, N}$, δ_i – качество входных данных; δ^{X_i} – допустимое качество

входных данных; θ_j – качество выходных данных; θ^{Y_j} – допустимое качество

выходных данных; T_t – текущее требование к системе; I, J – общее кол-во i -требований класса j ; $\Phi : \Phi_1, \Phi_2, \Phi_3, \Phi_4$ – множество функций системы;

Z – множество состояний системы, при которых достигается требуемый уровень защиты данных: обеспечение Φ_3 – функции защиты особых данных (z_i – i -е состояние системы; z^* – необходимое состояние системы после реализации i -го требования к системе в текущем состоянии; Z_w – множество верхних граней множества W , равных или больших всех элементов W), это утверждение может быть записано в следующей форме:

$$z^* = \sup(W) \mid z_i \in Z_w, \forall s \in Z_w : z^* \leq s,$$

Описание функции цели может быть определено по следующей схеме:

Пусть $F(T_j^{\Phi_i})$ – функционал, определенный на множестве $T_j^{\Phi_i}(X)$.

Рассматривается следующая задача нахождения варианта системы X :

$$x \in \arg \max_{x \in D} F(x); \arg \max_{x \in D} F(x) = \{x \in D \mid F(x) = \max_{x \in D} F(x)\}, \quad (3)$$

где $F(x)$ – искомый функционал.

Вариант $x \in D$, который максимизирует функционал $F(x)$, назовем оптимальным. Вариант x , при котором достигается максимум функционала $F(x)$ на некотором множестве $X^{(s_0)} \subseteq X$, называют максимальным. Полученные максимальные варианты исследуются с помощью критериев оптимальности.

3. ХОД РАБОТЫ

1. Определить вариант задания (согласовать с ведущим преподавателем или руководителем НИР);
2. Изучить литературу и теоретические сведения к выполнению лабораторной

работы;

3. Исходя из технологии RAD, провести анализ требований;
4. Привести формальное описание системы в целом;
5. Предложить функцию цели для проектирования системы;
6. Сформировать и защитить отчет, содержащий все этапы исследования.

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое требование к системе?
2. Описать основные задачи технологии RAD.
3. Какие существуют подходы к формальному описанию систем?
4. Привести пример описания реальной системы, когда должны быть учтены обратные связи в операторах преобразования объектов.
5. Прочесте и объясните формулу: $x \in \arg \max_{x \in D} F(x)$

5. ВАРИАНТЫ ЗАДАНИЙ

Вариант задания может быть выбран из предложенного списка по последней цифре номера зачетной книжки либо выбирается студентом индивидуально, в соответствии с тематикой выпускной (квалификационной) магистерской работы по согласованию с её руководителем.

Вариант 0. Конструкторское бюро по проектированию сложных технических объектов. Использование унифицированной САПР. Данные сохраняются в БД. Важность формирования целостного изображения.

Вариант 1. Станция экологического контроля параметров на химическом предприятии. ИС должна функционировать в оперативном режиме работы, решения об уровне загрязнения принимает система, подтверждает оператор. Необходимость защиты данных для хранения и при передаче на другие объекты.

Вариант 2. Предприятие по выпуску пищевой продукции обеспечено автоматизированной системой управления. Основные этапы технологического процесса: передача объектов по конвейеру, при котором необходим контроль качества выполненных операций. ИС собирает информацию со специальных датчиков и обрабатывает ее, сравнивая показатели объектов на конвейере с контрольными, при несовпадении отбраковывает, снимая с конвейера. Данные сохраняются в БД.

Вариант 3. Предприятие интернет торговли. Информационная система управляет пользователями, обеспечивает доступ, регистрацию и возможность

скидок. Наличие конкурентов должно быть сопоставлено с возможностью защиты данных в режиме реального времени и их сохранности.

Вариант 4. Юридическая контора обслуживается ИС с двумя БД различного назначения (клиенты, их дела и нормативно-правовая информация). При этом БД связаны. БД клиент-дело особо секретна, БД справочная открыта и имеет доступ по всей юридической компании. Юристы, ведущие дела, могут оперативно добавлять информацию по делам по мобильной связи.

Вариант 5. Предприятие лечебно-диагностического характера. ИС клиентов, где ранятся процедуры, описание диагнозов и процесса лечения. Поступление данных в БД из различных источников: датчики и устройства (напрямую с ультразвукового аппарата, от мангитно - резонансной диагностики, носимых устройств: например, пульс, частота дыхания, аритмия). Необходимость быстрого реагирования на особые ситуации в текущем режиме (информация с носимых устройств) и при диагностике.

Вариант 6. Администрация учебного заведения. ИС «школьный дневник» с возможностью обмена сообщениями с классным руководителем родителями учащегося при нестандартных ситуациях в оперативном режиме. Поступление данных с различных источников: проходная школы, журнал оценок, столовая, медицинский кабинет. Возможность сохранения и анализа данных за неделю, месяц, год.

Вариант 7. Предприятие контроля параметров технических устройств. Датчики и средства контроля передают комплекс информации о контролируемом устройстве в БД, откуда происходит их извлечение и анализ. Далее, на основании анализа формируется отчет о состоянии устройства. Возможность реализации контроля в скоростном режиме.

Вариант 8. Бюро технической инвентаризации (БТИ). Данные о состоянии жилого фонда вносятся как вручную, так и с помощью устройств фотосъемки, аудиозаписи (например, при определении шума при изоляции квартиры). Возможность передачи данных по локальной сети между подразделениями БТИ. Совместимость с другими БД.

Вариант 9. Станция гидрологического контроля параметров в морском порту. ИС должна функционировать в оперативном режиме работы, требуется наличие совместимых форматов с другими БД подобного профиля. Контроль параметров осуществляется как автоматически, так и вручную. Возможность предупреждения об опасных явлениях (резкое повышение уровня моря, увеличение скорости течения).

6. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Правила составления Software requirements specification [Электронный ресурс] /Режим доступа: <http://habrahabr.ru/blogs/development/52681/>. – Загл.с экрана.
2. Рамбо Дж. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо. – McMillan, 2007.
3. Резник А. Развивающиеся системы [Электронный ресурс] / А. Резник// International Conference «Knowledge-Dialogue-Solutions», 2007. – Режим доступа: <http://www.foibg.com/conf/ITA2007/KDS2007/PDF/KDS07-Reznik1.pdf>. – Загл. с экрана.
4. Дабагян А.В. Принципы автоматизированного проектирования систем машин и техноологических процессов: Учеб.пособие. – Харьков:ХПИ, 1987. – 66 с.
5. Дагабян А.В. Проектирование технических систем. – М.: Машиностроение, 1986. – 256 с.
6. Петров В.Н. Информационные системы / В.Н. Петров. – СПб.: Питер, 2003. – 688 с.

Требования к содержанию и оформлению отчетов

Отчеты по лабораторным работам и практическим заданиям оформляются согласно правилам оформления принятыми на кафедре, ГОСТам и ЕСКД.

Основные правила по оформлению отчетной документации:

Параметры страницы: А4 (21×29,7), ориентация – книжная (допускается использовать альбомную ориентацию страницы для выполнения схем и таблиц).

Поля: левое – 2.5, верхнее – 1.5, нижнее – 1.5, правое – 1.

Нумерация страницы – внизу, справа. Нумерация ведется с титульного листа, номер на титульном листе не ставиться.

Шрифт Times New Roman, кегль 14, интервал – одинарный.

Заголовки разделов: абзацный отступ – 0, выравнивание по центру, шрифт – жирный, нумерация – арабскими цифрами, точка в конце названия раздела не ставиться.

Заголовки подразделов (допускается три уровня, например 1.1., 1.1.1.): абзацный отступ – 1.25÷1.5, выравнивание по ширине, шрифт – жирный, точка в конце названия подраздела не ставиться.

Основной текст: абзацный отступ – 1.25÷1.5, выравнивание по ширине, шрифт – обычный.

Нумерация рисунков и таблиц – сквозная внутри раздела (например, в разделе 1 – рис. 1.1., рис.1.1.2 и т.д., или табл.1.1., табл.1.2. и т.д.).

Рисунки помещаются после упоминания их в тексте и имеют подпись, размещаемую под рисунком без абзацного отступа и имеющую выравнивание по центру и точку на конце названия (например, Рис.1.1. Название.).

Таблицы размещаются после ссылки на них в тексте. Название приводится над таблицей, без абзацного отступа с выравниванием по центру, без точки на конце названия (например (Таблица 2.2. Название)).

Допускается выносить рисунки и таблицы в Приложения. В этом случае ссылка должна содержать номер приложения (например: рис.1.1. Приложения 1 или табл.А1 Приложения А).

Основная часть должна содержать ссылки на используемую литературу или информационные источники, список которых приводится после раздела Выводы и перед Приложениями. Ссылка заключается в квадратные скобки (например – [1], [5,7], [3-6]).

Приложения нумеруются арабскими цифрами (Приложение 1, Приложение 2) или обозначаются русскими заглавными буквами в порядке их следования (Приложение А, Приложение Б). Слово Приложение....выравнивается по правому краю и имеет жирный шрифт. Название приложения располагается на следующей строке, без абзацного отступа, выравнивание по центру, шрифт – жирный.

По завершению изучения курса у студента должен быть сформирован набор отчетов (Приложение №2), сведенных в единый

документ и имеющий единый титульный лист (Приложение №3), на котором отражаются результаты прохождения этапов изучения дисциплины.

Каждый раздел этого документа является отчетом по выполнению соответствующего практического задания и лабораторной работы (обязательные разделы и правила выполнения отчетов представлены в Приложении 2).

Сформированный документ, с отметками о выполнении всех практических заданий и лабораторных работ обязателен для представления и является подтверждением о допуске к итоговому контролю.

Организация защиты и критерии оценивания выполнения лабораторных работ

К защите представляется отчет, включающий в себя результаты выполнения практического задания и лабораторной работы, выполненные согласно правилам, описанным в п.4.2 и единый титульный лист, на котором отмечаются результаты выполнения заданий.

На проверку теоретической подготовки, проводимой по контрольным вопросам, отводится 5-6 минут.

Степень усвоения теоретического материала оценивается по следующим критериям:

- **оценка «отлично» выставляется, если:**
 - последовательно, четко, связно, обоснованно и безошибочно с использованием принятой терминологии изложен учебный материал, выделены главные положения, ответ подтвержден конкретными примерами, фактами;
 - самостоятельно и аргументировано сделан анализ, обобщение, выводы, установлены межпредметные (на основе ранее приобретенных знаний) и внутрипредметные связи, творчески применены полученные знания в незнакомой ситуации;
 - самостоятельно и рационально используются справочные материалы, учебники, дополнительная литература, первоисточники; применяется систему условных обозначений при ведении записей, сопровождающих ответ; используются для доказательства выводы из наблюдений и опытов, ответ подтверждается конкретными примерами;
 - допускает не более одного недочета, который легко исправляется по требованию преподавателя.
- **оценка «хорошо» ставится, если:**
 - дан полный и правильный ответ на основе изученных теорий; допущены незначительные ошибки и недочеты при воспроизведении изученного материала, определения понятий, неточности при

использовании научных терминов или в выводах и обобщениях из наблюдений и опытов; материал излагает в определенной логической последовательности;

- самостоятельно выделены главные положения в изученном материале; на основании фактов и примеров проведено обобщение, сделаны выводы, установлены внутрипредметные связи.
- допущены одна негрубая ошибка или не более двух недочетов, которые исправлены самостоятельно при требовании или при небольшой помощи преподавателя; в основном усвоил учебный материал.

• ***оценка «удовлетворительно» ставится, если:***

- усвоено основное содержание учебного материала, но имеются пробелы в усвоении материала, не препятствующие дальнейшему изучению; материал излагает несистематизированно, фрагментарно, не всегда последовательно;
- показана недостаточная сформированность отдельных знаний и умений; выводы и обобщения аргументируются слабо, в них допускаются ошибки;
- допущены ошибки и неточности в использовании научной терминологии, даются недостаточно четкие определения понятий; в качестве доказательства не используются выводы и обобщения из наблюдений, фактов, опытов или допущены ошибки при их изложении;
- обнаруживается недостаточное понимание отдельных положений при воспроизведении текста учебника (записей, первоисточников) или неполные ответы на вопросы преподавателя, с допущением одной – двух грубых ошибок.

• ***оценка «неудовлетворительно» ставится, если:***

- не усвоено и не раскрыто основное содержание материала; не сделаны выводы и обобщения;
- не показано знание и понимание значительной или основной части изученного материала в пределах поставленных вопросов или показаны слабо сформированные и неполные знания и неумение применять их к решению конкретных вопросов и задач по образцу;
- при ответе (на один вопрос) допускается более двух грубых ошибок, которые не могут быть исправлены даже при помощи преподавателя;
- не даются ответы ни на один из поставленных вопросов.

Оценка выполнения практических заданий и лабораторных работ проводится по следующим критериям

• ***оценка «отлично» ставится, если студент:***

- творчески планирует выполнение работы;
- самостоятельно и полностью использует знания программного материала;
- правильно и аккуратно выполняет задание;
- умеет пользоваться литературой и различными информационными источниками;

- выполнил работу без ошибок и недочетов или допустил не более одного недочета

- ***оценка «хорошо» ставится, если студент:***

- правильно планирует выполнение работы;
- самостоятельно использует знания программного материала;
- в основном правильно и аккуратно выполняет задание;
- умеет пользоваться литературой и различными информационными источниками;
- выполнил работу полностью, но допустил в ней: не более одной негрубой ошибки и одного недочета или не более двух недочетов.

- ***оценка «удовлетворительно» ставится, если студент:***

- допускает ошибки при планировании выполнения работы;
- не может самостоятельно использовать значительную часть знаний программного материала;
- допускает ошибки и неаккуратно выполняет задание;
- затрудняется самостоятельно использовать литературу и информационные источники;
- правильно выполнил не менее половины работы или допустил:
 - не более двух грубых ошибок или не более одной грубой и одной негрубой ошибки и одного недочета;
 - не более двух- трех негрубых ошибок или одной негрубой ошибки и трех недочетов;
 - при отсутствии ошибок, но при наличии четырех-пяти недочетов.

- ***оценка «неудовлетворительно» ставится, если студент:***

- не может правильно спланировать выполнение работы;
- не может использовать знания программного материала;
- допускает грубые ошибки и неаккуратно выполняет задание;
- не может самостоятельно использовать литературу и информационные источники;
- допустил число ошибок недочетов, превышающее норму, при которой может быть выставлена оценка «3»;
- если правильно выполнил менее половины работы;
- не приступил к выполнению работы;
- правильно выполнил не более 10% всех заданий.

Список экзаменационных вопросов
по дисциплине Системная инженерия
для студентов направления 09.04.02 – “Информационные системы и
технологии” всех форм обучения

1. Понятие системной инженерии. Общие принципы системной инженерии.
2. Этапы развития СИ. Связь системной инженерии и системотехники.
3. Практики СИ. Понятие архитектуры и архитектурной деятельности.
4. Проблема унификации при построении сложной системы: языки, формы описания организационного процесса и структурных единиц.
5. Объект и субъект коммуникации в СИ. Роль и компетенции системного архитектора.
6. Организации, связанные с СИ: INCOSE, VKCASE, CESUN.
7. Классификация основных стандартов на информационные системы и технологии.
8. Логическая архитектура и физическая архитектура систем в ISO 15288 (ГОСТ Р ИСО/МЭК 15288-2005).
9. Формат типового описания практики (ISO 24774): название, назначение, результаты, состав (мероприятия и дела).
10. Классификация сложных систем. Рубрикаторы, исследователи.
11. Подходы к формализации описания систем и процессов (функциональный, структурный, процессный).
12. Формализация информационных процессов в сложной системе.
13. Формализация связей, описание структур и функций сложных систем. Квантируемость пространства и времени.
14. Формализация жизненного цикла (ЖЦ) сложной системы. Наложение циклов. Раздельный анализ ЖЦ проектирования и функционирования ИС.
15. Функциональное описание систем. Макроскопическое представление.
16. Системное окружение. Надсистемы.
17. Системотехнический подход к построению моделей функционирования сложных систем.
18. Формализованное описание системы в целом и отдельного элемента.
19. Формализованное описание свойств системы.
20. Уравнение состояния системы. Свойства операторов G и H.
21. Классификация системных моделей.
22. Модели CTMF, CMF.
23. Модели DMF, DLMF.
24. Модели MF, NMF.
25. Модели NLMF, LMF.
26. Синтез систем на элементах с ограниченной сочетаемостью.
27. Информационные характеристики сложных систем.
28. Прагматические характеристики качества информации в сложной системе. Формализация показателя значимости информации.
29. Схема формирования информации в системе.

30. Уровни изучения систем. Определение границ системы.
31. Методы описания структур. Структурная схема двухуровневой системы управления сложной системой.
32. Оценка реализуемости элементов структуры.
33. Морфологический синтез систем.
34. Методы генерации вариантов структурных компонент сложных систем.
35. Методы внешнего проектирования систем: вертикальные («сверху вниз», «снизу вверх»), горизонтальные («кубики», метод прототипов).
36. Топологический анализ структур. Диаметр структуры, степень централизации структуры системы.
37. Определение показателя сложности системы. Принцип монотонности систем.
38. Системосоздающие и системоразрушающие факторы. Примеры систем с различными факторами.
39. Организация и система. Сложность, степень дифференциации, лабильность.
40. Проблема многокритериальности при оценке сложных систем.
41. Модели качества: ISO9126, ГОСТ 34, по МакКоллу, по Бозму, 1061-1998 IEEE, ISO 8402:199.
42. Инженерия требований к сложной системе. ГОСТ 34 (разделы 4-8).

Приложение 1

Образец оформления и содержания отчета по лабораторной работе Титульный лист

Министерство образования и науки Российской Федерации ФГАОУ ВО «Севастопольский государственный университет» Институт информационных технологий и управления в технических системах			
Кафедра Информационные системы			
Отчет по лабораторному практикуму по дисциплине «Системная инженерия»			
Л.р. №	Дата защиты	Оценка	Подпись преподавателя
1			
2			
3			
4			
Итого			
Выполнил: ст. гр. <u>ИС/м-XX-о(з)</u> <i>ФИО студента</i> Проверил: <i>ФИО преподавателя</i>			
Севастополь 20__			

Содержание отчета по лабораторной работе Лабораторная работа №__

Тема:

Цель:

Основная часть

Отчет о выполнении этапов задания на лабораторную работу; распечатки программ и результатов их работы; схемы и графики.

Выводы

Примерный текст: *«В процессе реализации исследований по теме... были применены методы... алгоритмы... подходы..., которые позволили.... получить следующие основные результаты... Особенностью реализации исследования для заданной предметной области явилось...В приложении данного исследования для текущей НИР...были получены следующие результаты.... »*