

ОЦЕНКА ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

А.М. Зубаха, кандидат технических наук;

Т.А. Подружкина, кандидат педагогических наук.

Санкт-Петербургский университет ГПС МЧС России

Описаны подходы к оценке эффективности автоматизированных систем управления, представлены показатели и математические методы их расчета, рассмотрены примеры расчета показателей эффективности информационных автоматизированных систем управления.

Ключевые слова: автоматизированная система управления, показатели эффективности, человеко-машинная система, алгоритм функционирования, вероятность достижения поставленной цели, время достижения поставленной цели, надежность программного обеспечения

EVALUATION OF THE EFFECTIVENESS OF AUTOMATED INFORMATION CONTROL SYSTEMS

A.M. Zubaha, T.A. Podrzhkina.

Saint-Petersburg university of State fire service of EMERCOM of Russia

The approaches to evaluating the effectiveness of automated control systems, and presents the mathematical methods of their calculation, consider the example of calculating the efficiency of information control systems.

Keywords: automated control system, performance indicators, human-machine system, algorithm of functioning, probability of achieving the objectives, time to achieve this goal, reliability of the software

Автоматизированная система управления (АСУ) – это человеко-машинная система (ЧМС), представляющая собой комплекс аппаратных и программных средств, предназначенный для управления различными организационными и технологическими процессами.

Современные автоматизированные системы строятся, как правило, на основе компьютерных сетей. Компьютерная сеть в совокупности с программным обеспечением и пользователями, по сути, и есть человеко-машинная система, призванная повысить эффективность, прежде всего управленческой деятельности человека. Поэтому при оценке эффективности АСУ необходимо учитывать такие составляющие управленческого процесса, как надежность и оперативность функционирования системы [1].


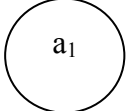
В качестве основных показателей эффективности предлагаются следующие показатели:

- вероятность достижения поставленной цели ($P_{ц}$);
- время достижения поставленной цели ($M(T_{ц})$);
- надежность программного обеспечения (ПО).

Для оценки вероятности достижения поставленной цели и времени достижения поставленной цели необходимо построить алгоритм функционирования АСУ при выполнении управленческих действий. Алгоритм функционирования удобно представить в виде сетевого графика и поясняющей таблицы [2–4].

В сетевом графике и таблице приняты следующие обозначения (табл. 1).

Таблица 1

Обозначение	Назначение
A_{1-3} 	Стрелкой обозначается дуга сетевого графика, обозначающая выполняемую работу между двумя узлами для перехода ЧМС из одного состояния в другое
	Узел сетевого графика (вершина), обозначающий начальное состояние ЧМС или после выполненной конкретной работы

Применение сетевого графика иллюстрируется на примере построения упрощенного алгоритма функционирования Центра управления кризисными ситуациями (ЦУКС) [5, 6] при поступлении сигнала о чрезвычайной ситуации (рис. 1, табл. 2).

Таблица 2

№ п/п	Работа	Содержание работы	Исполнитель (ч – человек, м – машина)	Вершина, на которую опирается работа
1.	A_{1-2}	Поступление сигнал в ЦУКС о возникновении ЧС	ч	a_1
2.	A_{2-3}	Оценка обстановки	ч, м	a_2
3.	A_{3-4}	Направление сил МЧС России на ликвидацию ЧС	ч,	a_3
4.	A_{4-5}	Прибытие сил МЧС России к месту возникновения ЧС	ч	a_4
5.	A_{5-10}	Начало борьбы с ЧС	ч	a_5
6.	A_{5-6}	Уточнение обстановки на месте и доклад в ЦУКС	ч	a_5
7.	A_{6-7}	Анализ в ЦУКС обстановки	ч, м	a_6
8.	A_{7-8}	Выработка дополнительных рекомендаций по борьбе с ЧС	ч, м	a_7
9.	A_{8-9}	Направление дополнительных сил МЧС России на ликвидацию ЧС	ч	a_8
10.	A_{9-10}	Продолжение борьбы с ЧС	ч	a_9
11.	A_{10-11}	Ликвидация ЧС и доклад в ЦУКС	ч, м	a_{10}
12.	A_{11-12}	Анализ	ч, м	a_{11}
13.	A_{12-13}	Подготовка отчета	ч, м	a_{12}

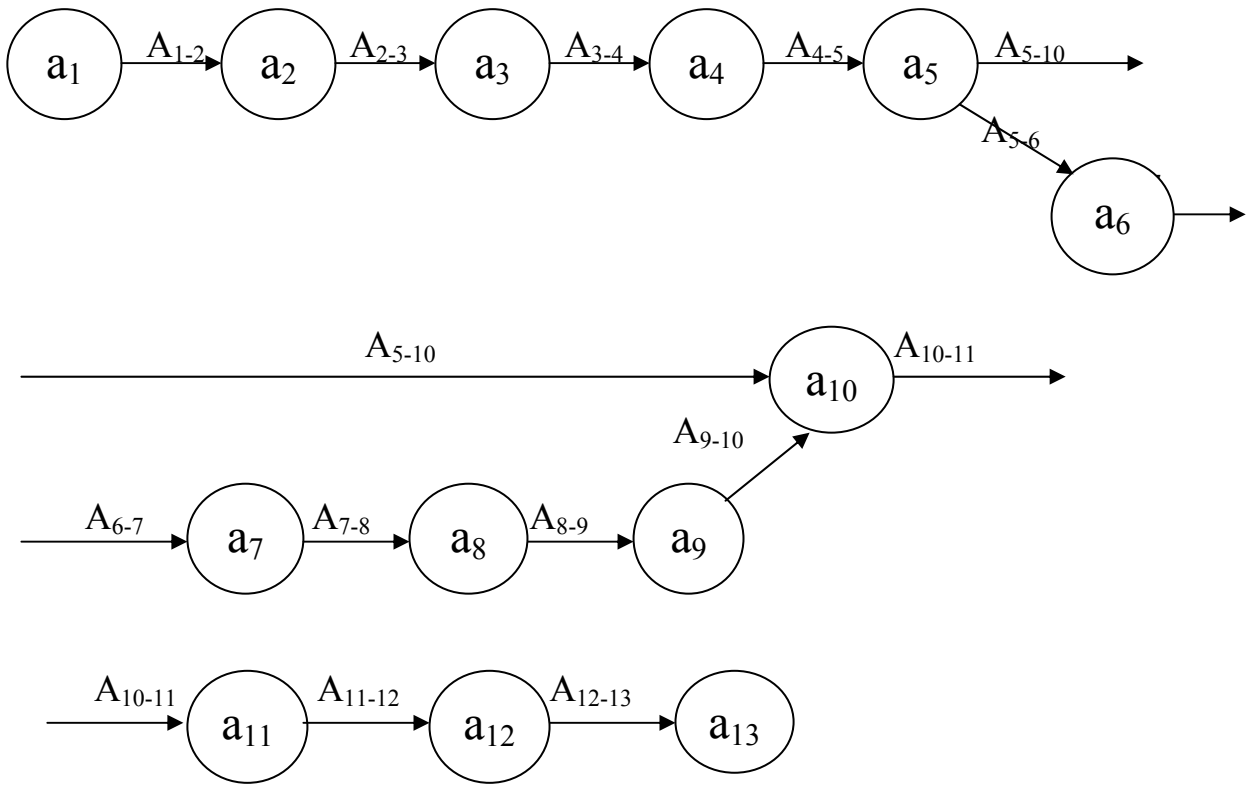


Рис. 1. Алгоритм функционирования

Вероятность достижения поставленной цели (P_u)

По разработанному алгоритму функционирования системы управления проводится большое количество реализаций, и вероятность достижения поставленной цели определяется соотношением [3, 6]:

$$P_u = \frac{m}{n},$$

где n – общее число реализаций алгоритма функционирования; m – число положительных реализаций алгоритма функционирования.

Реализация алгоритма функционирования считается положительной, если все A_i его работы выполнены и если хотя бы одна работа не выполнена – реализация считается отрицательной.

Вероятность одной P_{ui} реализации определяется выражением:

$$P_{ui} = \prod_{i=1}^k P(A_i),$$

где $P(A_i)$ – вероятность выполнения A_i работы; i – номер работы; k – общее количество работ.

При отрицательной реализации алгоритма функционирования вероятность невыполнения A_i работы или выполнения работы с ошибкой:

$$P'(A_i) = 1 - P(A_i).$$

Выбор положительной или отрицательной реализации A_i определяется на основании статистических данных работы ЧМС, полученных на этапе эксплуатации системы.

Время достижения поставленной цели ($M(T_u)$)

Время достижения поставленной цели определяется как математическое ожидание времени выполнения одной реализации алгоритма функционирования:

$$M(T_u) = \sum_{i=1}^m M(T_i),$$

где $M(T_i)$ – математическое ожидание времени выполнения A_i работы.

В свою очередь:

$$M(T_i) = (\sum_{i=1}^m T_i) / m,$$

где m – число положительных реализаций алгоритма функционирования; T_i – время выполнения i -ой работы.

Из приведенных формул видно, что при определении математического ожидания времени выполнения учитываются только положительные реализации алгоритма функционирования.

Если известны надежность и время выполнения работы в условиях отсутствия компьютерной сети, то тогда эффективность внедрения сетей определится с помощью следующих показателей: приращения надежности функционирования ЧМС за счет внедрения сети (ΔP_c) и приращения оперативности функционирования ЧМС за счет внедрения сети (ΔT_c):

$$\Delta P_c = P_{ц} - P_{бс};$$

$$\Delta T_c = M(T_{ц}) - M(T_{бс}).$$

Надежность программного обеспечения

Надежность ПО характеризуется его безотказностью и восстанавливаемостью.

Безотказность ПО – это свойство сохранять свою работоспособность при использовании его в процессе функционирования ЭВМ.

Безотказность ПО целиком определяется его корректностью, то есть зависит от наличия ошибок, внесенных в него на этапе создания, и эта зависимость, отражающая изменение во времени надежности ПО, имеет вид, представленный на рис. 2 [3, 4].

Интенсивность отказов

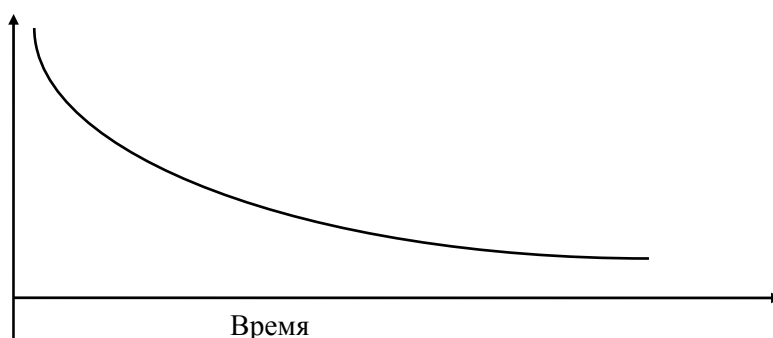


Рис. 2. Интенсивность отказов ПО

Восстанавливаемость ПО определяется затратами времени и труда на устранение отказа, возникшего из-за появившейся программной ошибки, и может быть оценена средней продолжительностью устранения ошибки в программе и восстановления ее работоспособности.

Причины отказов ПО:

1. Ошибки, скрытые в самой программе.

В процессе создания программного продукта невозможно обнаружить и ликвидировать все ошибки. Наличие скрытых ошибок – главный фактор нарушения нормальных условий функционирования ПО. Выделяют следующие классы ошибок в программах:

- ошибки вычислений;
- логические ошибки;
- ошибки ввода-вывода;
- ошибки манипулирования данными;
- ошибки совместимости;
- ошибки сопряжений.

2. Искажение входной информации, подлежащей обработке.

Возникает тогда, когда входные данные не подаются в область допустимых значений переменных программы. Причинами искажения входной информации являются:

- искажение данных на первичных носителях информации;
- сбои и отказы в аппаратуре передачи или приема информации;
- сбои в канале связи;
- ошибки пользователя при подготовке входной информации.

3. Неверные действия оператора.

4. Неисправности аппаратуры.

В зависимости от степени серьезности последствий ошибок в программе может возникнуть полное прекращение выполнения функций на длительное или неопределенное время или краткосрочное нарушение вычислительного процесса. Наиболее типичными признаками появления ошибок в программе являются:

- преждевременное (аварийное) окончание выполнения программы;
- недопустимое увеличение времени выполнения программы или ее закливание;
- полная потеря или значительное искажение накопленных данных, необходимых для решаемых задач или искажение отдельных элементов данных в результате обработки искаженной исходной информации;
- нарушение последовательности вызова отдельных программ или отдельных модулей программы.

Для оценки надежности ПО используются аналитические модели надежности, которые дают возможность исследовать закономерности появления ошибок в программах, а также прогнозировать надежность ПО при его разработке и эксплуатации. Эти модели строятся на предположении, что появление ошибки является случайным событием и, следовательно, имеет вероятностный характер. Поэтому при построении моделей используются следующие характеристики надежности:

- функция надежности $P(t)$, определяемая как вероятность того, что ошибки программы не проявятся на интервале времени от 0 до t , то есть время ее безотказной работы будет больше t ;
- функция ненадежности $Q(t)$, определяется как вероятность того, что на интервале времени от 0 до t возникнет отказ ПО.

Таким образом, из приведенных определений следует:

$$Q(t) = 1 - P(t).$$

Средняя наработка на отказ T_0 – математическое ожидание временного интервала между последовательными отказами.

Интенсивность отказов $\lambda(t)$ – определяется как среднее число отказов в единицу времени. Интенсивность отказов и функция надежности связаны между собой выражением:

$$P(t) = \exp[-n\lambda(t)dt].$$

Число обнаруженных отказов (n) – определяется как количество выявленных отказов на интервале тестирования.

Как было сказано выше, оценка показателей надежности производится на основе аналитических моделей надежности ПО, которых существует большое множество:

- модель надежности программ с дискретно-понижающейся частотой (интенсивностью) появления ошибок;
- модель надежности программ с дискретным увеличением времени наработки на отказ;
- экспоненциальная модель надежности программ;
- модель надежности больших программных комплексов;
- большое количество моделей, названных по имени их авторов (Джелинского-Моранды, Шика-Уолвертона, Липова и др.).

Ниже дана характеристика экспоненциальной модели и модели оценки надежности больших программных комплексов.

Экспоненциальная модель основана на предположении об экспоненциальном характере изменения во времени числа ошибок в программе и на предположении о том, что интенсивность отказов пропорциональна числу оставшихся ошибок. В качестве показателей надежности ПО в модели используются:

- T – среднее время наработки на отказ;
- n – число обнаруженных отказов;
- $\lambda(\tau)$ – интенсивность отказов.

Пусть: M – число ошибок в программе перед фазой тестирования; $m(\tau)$ – конечное число исправленных ошибок; $m_0(\tau)$ – число оставшихся ошибок; τ – интервал тестирования (время тестирования).

Тогда:

$$m_0(\tau) = M - m(\tau).$$

Если исходить из предположения, что в процессе корректировки новые ошибки не появляются, то есть интенсивность исправления ошибок равна интенсивности их обнаружения, то в этом случае среднее время наработки на отказ (T_0) и интенсивность наработки на отказ определяются выражениями:

$$T_0 = T_0' \exp(\lambda(\tau) / MT_0');$$

$$\lambda(\tau) = 1/T_0,$$

где T_0' – исходное значение среднего времени наработки на отказ перед тестированием.

Очевидно, что время наработки на отказ увеличивается по мере выявления и устранения ошибок.

Оценка надежности больших программных комплексов трудоемкий процесс, связанный с большим количеством тестовых прогонов всего комплекса. Показателем надежности больших программных комплексов является вероятность безотказной работы (P_k) и определяется она следующим соотношением:

$$P_k = m/n,$$

где n – общее количество прогонов; m – количество прогонов с положительным результатом.

Большие программные комплексы, как правило, строятся по модульному принципу, поэтому надежность каждого n -го прогона всего программного комплекса определяется как функция надежности его составных частей:

$$P_{kn} = f(P_{ki}),$$

где P_{kn} – вероятность безотказной работы всего программного комплекса за один прогон; P_{ki} – вероятность безотказной работы i -го программного модуля.

При определении P_{kn} приняты следующие допущения:

- надежности модулей программного комплекса величины независимые;
- общий результат работы программы будет некорректным, если хотя бы в одном модуле обнаружена в процессе прогона ошибка;
- надежность каждого модуля определена;
- надежность межмодульного обмена определена.

Функционирование программного комплекса описывается направленным графом состояний (рис. 3). На представленном графе каждая вершина, обозначенная кругом, соответствует программному модулю, который имеет определенную надежность (P_{ki}), а каждая дуга соответствует направлению передачи управления от модуля к модулю. Передача управления от модуля к модулю также осуществляется с определенной надежностью (P_{ij}).

Таким образом, показателями надежности большого программного комплекса при n прогоне являются названные выше: вероятность безотказной работы всего за один прогон программного комплекса, вероятность безотказной работы i -го программного модуля и P_{ij} – вероятность безотказной передачи управления между модулями.

Если модуль N_i выдает правильный результат, то осуществляется переход к выполнению модуля N_j с вероятностью $P_{ki}P_{ij}$. Переход в состояние С соответствует корректному завершению программы и происходит с вероятностью (P_{kn}). Эта вероятность определяется следующим выражением:

$$P_{kn} = \prod_{i=1}^N P_{ki}P_{ij}.$$

При появлении ошибки переход осуществляется в состояние F, что соответствует некорректному выполнению программы, независимо от того как выполнены другие модули программного комплекса и, как следствие отрицательной реализации прогона в целом.

Рассмотрим на примере.

Программный комплекс состоит из семи модулей и его функционирование описывается графом состояний, представленным на рис. 2. Требуется определить:

1. Вероятность безотказной работы программного комплекса за 55 прогонов, из которых три было отрицательных.

2. Вероятность безотказной работы программного комплекса за один прогон, если $P_{k1}=0,992$; $P_{k2}=0,995$; $P_{k3}=0,989$; $P_{k4}=0,997$; $P_{k5}=0,99$; $P_{k6}=0,987$; $P_{k7}=0,999$; $P_{1-2}=0,991$; $P_{1-3}=0,993$; $P_{2-4}=0,998$; $P_{2-5}=0,989$; $P_{3-6}=0,996$; $P_{4-6}=0,997$; $P_{4-7}=0,995$; $P_{5-7}=0,988$; $P_{6-7}=0,993$; $P_{7-c}=1$.

1. $P_k = 52/55 = 0,945$.

2. $P_{kn} = (0,992 * 0,991) * (0,992 * 0,993) * (0,995 * 0,998) * (0,995 * 0,989) * (0,989 * 0,996) * (0,997 * 0,997) * (0,997 * 0,995) * (0,99 * 0,988) * (0,987 * 0,993) * 0,999 = 0,88$.

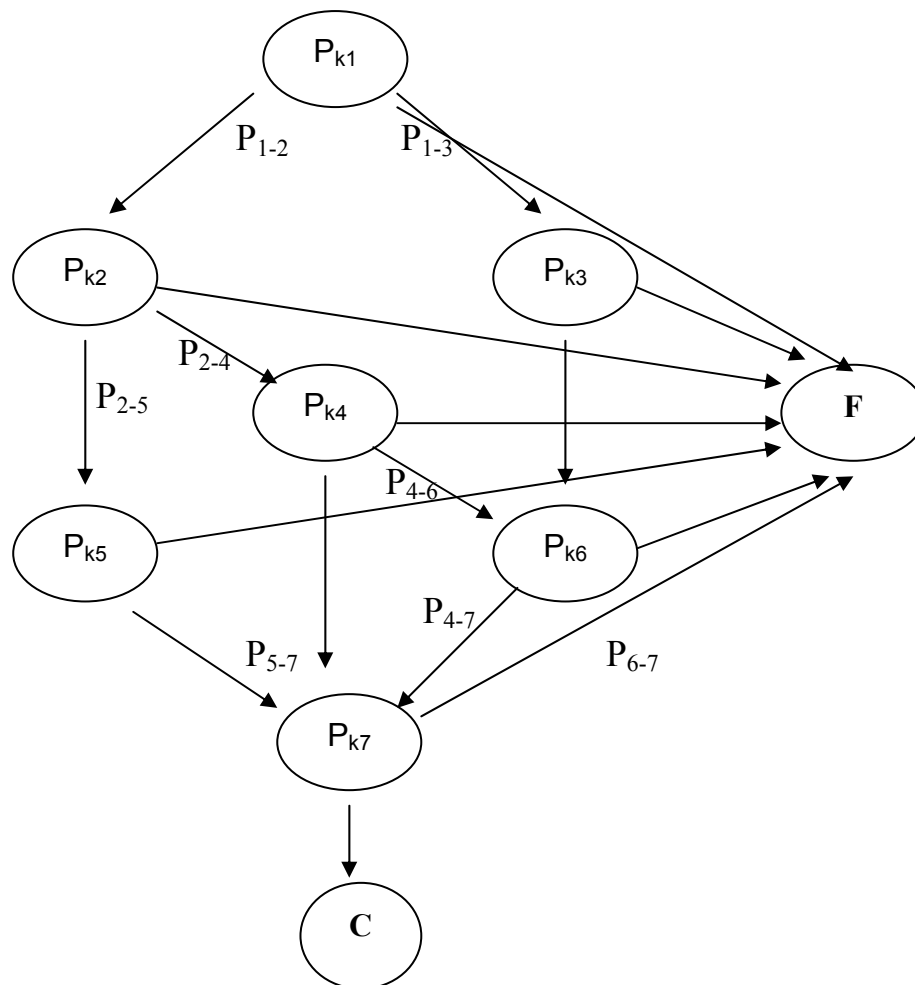


Рис. 3. Граф состояний программного комплекса

Литература

1. Давыдов А.В., Куклин В.В., Хайдаров К.А. Информационные сети и телекоммуникационные каналы. URL: <http://bourabai.kz/telecom/index.htm> (дата обращения: 20.05.2014).
2. Балдин К.В., Воробьев С.Н., Уткин В.Б. Управленческие решения: учеб. 5-е изд. М.: Изд-ко-торговая корпорация «Дашков и Ко», 2008. 496 с.
3. Шпак В.Ф. Основы автоматизации управления. Петродворец: Воен.-морской ин-т радиоэл. Им. А.С. Попова, 1998. Ч. 1. С. 174–181.
4. Автоматизация управления и связь в ВМФ / Н.Ф. Директоров [и др.]. СПб.: Элмор, 2001. С. 176–179.
5. Голубков Е.П. Технология принятия управленческих решений. М.: Дело и сервис, 2005.
6. Вентцель Е.С. Исследование операций. М.: Высш. шк., 2007.