

---

## Критерии профессиональности программистов при разработке экспертной системы управления программными проектами

*А.А. Живаева, В.В. Долгов*

*Донской Государственный Технический Университет, Ростов-на-Дону*

**Аннотация:** В течение долгого времени в программной инженерии существует проблема, связанная с довольно большим процентом провалов программных проектов. Одной из наиболее частых причин таких провалов является неоптимальное распределение людских ресурсов, т.е. распределение разработчиков по задачам в соответствии с их навыками и спецификой задачи. Сфера разработки программного обеспечения требует постоянного совершенствования навыков, причем степень овладения одним навыком может положительно влиять на другие, связанные с ним. Из-за большого числа навыков существует потребность в экспертной системе, позволяющей отслеживать изменяющиеся навыки разработчиков и осуществлять близкое к оптимальному распределение людских ресурсов. При этом из-за невозможности однозначного оценивания навыков и частичного их перекрытия друг с другом, обработка информации о сотрудниках более удобна с использованием теории нечетких множеств, нежели при применении классической теории множеств или теории вероятности. В конце статьи приведен список наиболее значимых навыков при реализации ИТ-проектов.

**Ключевые слова:** экспертные системы, разработка программ, управление проектами, распределение людских ресурсов, навыки, нечеткие множества, неопределенность, теория вероятности.

На протяжении долгого времени проблема разработки программного обеспечения не утрачивает свою актуальность. Научно-исследовательская организация Standish Group International, Inc с 1994 года предоставляет статистику успешности программных проектов по всему миру. В 2015 году из 50 000 проектов 19% полностью провалились, т.е. не завершились вовсе [1]. При этом за предыдущие пять лет процент провалов варьировался от 17% до 22%, что говорит о закономерности данной проблемы в программной инженерии. Также эта организация предоставляет процентное выражение влияния различных факторов на успешность проекта. Согласно данным 45% успеха зависит от руководителей и команды разработчиков. Таким образом, можно сделать вывод, что планирование и управление проектами зачастую осуществляется не должным образом, в связи с чем имеется необходимость в создании системы поддержки принятия решений

при управлении процессом разработки программного обеспечения. Существующие разработки в данной области могут с разной степенью помочь в разрешении имеющейся проблемы, но они не позволяют осуществлять близкое к оптимальному распределение разработчиков по задачам проекта в соответствии с их навыками [2].

Чтобы выполнить данное распределение сотрудников по задачам, менеджер проекта должен опираться на имеющиеся у сотрудников навыки и специфику задачи. Каждая задача требует наличия у разработчика некоторого множества навыков. Для того чтобы выполнить распределение людских ресурсов можно использовать теорию множеств. Тогда пусть  $U$  – некоторое множество всех возможных навыков.  $X$  является множеством навыков, которые требуются для решения поставленной задачи ( $X \subseteq U$ ).  $Y$  является множеством навыков, имеющихся у отдельного разработчика ( $Y \subseteq U$ ). Множество  $Y$  либо определяется самим сотрудником или менеджером проекта, либо рассчитывается на основе данных о предыдущем опыте. Разработчик, у которого  $Y \cup X = X$  является наиболее подходящей кандидатурой для данной задачи. В случае если таких разработчиков нет, наиболее подходящим будет являться тот, у которого имеется наибольшее количество требуемых для решения задачи навыков. Данное описание предполагает, что разработчик может обладать или не обладать каким-либо навыком. В связи с этим возможна следующая ситуация: разработчик с более низкой квалификацией в каком-либо навыке может посчитать ее вполне достаточной для обладания данным навыком, а разработчик с более высокой квалификацией может посчитать ее недостаточной для обладания данным навыком. При этом произойдет неоптимальное распределение сотрудников по задачам. Аксиома исключающего третьего, лежащая в основе теории множеств, накладывает некоторые ограничения на ее применение во многих реальных задачах, в которых используются оценки типа: «отлично», «очень хорошо», «хорошо», «удовлетворительно», «плохо», «очень плохо» и

---

т.п. [3]. Данные оценки не могут адекватно формализовываться в классической теории множеств.

Так же возможна интерпретация неопределенных оценок с помощью теории вероятности. Но при использовании данного аппарата неопределенность степени обладания навыком отождествляется со случайностью. То есть набор навыков конкретного разработчика в данном случае определяется вероятностью обладания определенным навыком. А навыки рассматриваются как совокупность статистически однородных случайных объектов.

Однако рассматриваемая неопределенность является неопределенностью другого рода и не связана со случайностью. Она связана с лингвистической неопределенностью или нечеткостью рассуждений и восприятий того или иного навыка. Для формализации неопределенностей такого рода удобно использовать лингвистические переменные из теории нечетких множеств, которую в 1965 году предложил в своей статье «Fuzzy Sets» профессор Л. Заде из университета Беркли. [4] В соответствие с этой теорией нечеткое множество  $Y$  – множество упорядоченных пар  $\{y, \mu_Y(y)\}$ , где  $\mu_Y(y)$  является функцией принадлежности  $y$  в  $Y$  ( $\mu_Y(y) \rightarrow [0, 1]$ ) [5], – можно сопоставить с каким-либо навыком разработчика, а его функцию принадлежности – со степенью владения конкретным разработчиком ( $y$ ) данного навыка. При этом значение функции принадлежности равное 0 будет означать абсолютное отсутствие навыка у разработчика, а значение 1 – полное и абсолютное знание разработчиком всех тонкостей навыка. С помощью данной теории можно хорошо интерпретировать субъективные оценки «отлично», «очень хорошо», «хорошо», «удовлетворительно», «плохо» и «очень плохо» выставляемые разработчику экспертами или руководителями проектов. Связки («и», «или», «не», «очень», «более» и т.д.) в этом случае трактуются как видоизменяющие операторы, что позволяет при разработке экспертной системы переводить

запросы, составленные с учетом лингвистических переменных, в SQL-операторы реляционных баз данных, что упростит создание такой системы в целом. [6]

Еще одним важным вопросом является вид функции принадлежности – функции владения разработчиком тем или иным навыком. Вид этой функции напрямую зависит от особенностей рассматриваемой области и важности различия между собой значений лингвистических переменных. В рассматриваемой нами области оценивания навыков разработчиков крайне важно отличить разработчика, владеющего навыком «плохо» от разработчика, владеющего навыком «отлично». В то время как различие между «очень плохо» и «плохо», а также между «отлично» и «безупречно» может быть не таким важным. Кроме того, в крайних областях значений (области «очень плохо» и «очень хорошо») разница между разработчиками может быть спорной или неочевидной. Приемлемой функцией оценивания при таких условиях можно считать сигмоид с общей формулой

$$\mu(y) = \frac{1}{1 + e^{-a(y-0.5)}}, \quad \text{где «a» – коэффициент крутизны функции}$$

принадлежности (рис. 1).

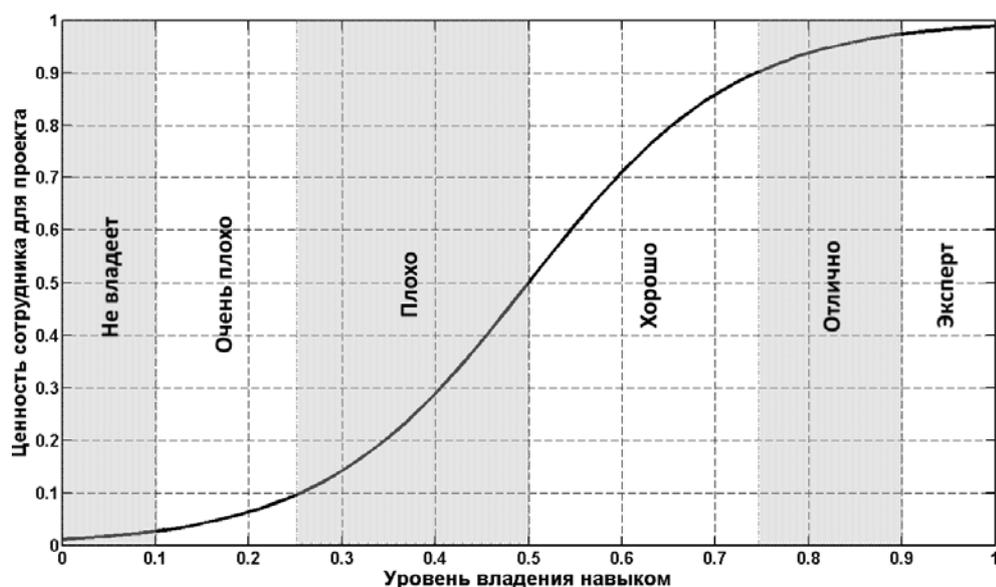


Рис. 1. – Функция ценности программиста от уровня владения навыком

Для того чтобы выполнять распределение людских ресурсов необходимо сформировать базу навыков, необходимых для выполнения различных задач. Множество всех возможных навыков представлено в таблицах 1-4 [7–12].

Таблица № 1

Навыки программирования в срезе языков программирования

Язык	Навыки	Язык	Навыки	Язык	Навыки
C	ACL	C++	ACE	Java	Apach Common Collections
	IUP		ACDK		Google Guava
	GSL		Boost		Jsoup
	SIGIL		STL		JDOM
	Imdb		Dlib		Gson
	ATTR		Loki		JasperReports
	CuTest		Reason		Hibarnate
	cUnit		C++ REST SDK		Commons Math
	Check		cppunit		Spring
	Cmocka		Google Test		Akka
JavaScript	jQuery	C#	jest		Arquillian
	jQuery UI		doctest		Jtest
	AngularJS		Windows Forms		Junit
	Angular2		WPF		Mockito
	React		LINQ		Powermock
	AJAX		PLINQ		The Grinder
	TypeScript		LINQ to SQL		perf4j
	Node.js		ASP.Net MVC		jetm
	ECMAScript5		WFC		JavaMelody
	ECMAScript6		.Net Core		Django
	Bootstrap	Moq	Requests		
	Underscore.js	FsCheck	wxPython		
	Echo	xUnit	NumPy		
	Vue.js	Nunit	PuQT		
	Jasmine	Bogus	Python-patterns		
	Mocha	Spark	Python-oauth2		
	Qunit	Hadoop	Pylint		
	Tape	Akka	multiprocessing		
	Unit.js	PostgreSQL async	Orange		
	Swift	OhMyAuth	Scala	MySQL async	Babel
SwiftCop		ReactiveMongo		PuSpark	
SwiftLine		Scala-redis		pytest	
Safe		ScalikeJDBC		PyAutoGUI	
Sync		Slick		mock	
SwiftData		Finatra		church	

	SwiftStructures		Play	Objective-C	Gtrack
	Algorithm		Scalatra		ocstyle
	SwiftTask		Spray		ObjectiveRecords
	Swiftx		SecureSocial		FMDB
	Swift-AI		json4s		RestKit
	SwiftMath		Pickling		FxForms
	RxSwift		PredictionIO		BFKit
	SwiftCheck		Kafka		CocoaPods
	Quick		Scala.Rx		Specta
	Fakery		Scalaz		Quick
	Perl		Catalyst		
Mojolicious		ScalaCheck	SQL PL		
Dancer		Specs2	SQL/PSM		
PSGI/Plack		ScalaTest	PL/SQL		
PerlUnit		Akka.Net	PL/pgSQL		
Test::Unit		Math.Net	Shiny		
SCTE		FsharpLu	Runit		
Test::Harness	canopy	RGG			
PHP	Symfony	F#	FsCheck	R	MilanoR
	Yii 2		FsUnit		dplyr
	Phalcon		xUnit.Net		broom
	CakePHP		Freya		SparkR
	PHPSpec		ASP.Net Core MVC		Rcpp
	PHPUtil		Nancy		Runit
	HTTP Mock		WevSharper		svUnit
	Faker				DSGeek
					rddr

Таблица № 2

### Навыки администрирования информационных систем

Область	Навыки	Область	
Администрирование баз данных	Управление БД	Администрирование веб-сервера	Знание Unix-систем
	Управление пропускной способностью БД		Конфигурация веб-серверов
	Создание БД		Конфигурация почтовых серверов
	Мониторинг БД и настройка производительности		Знание ОС семейства Windows Server
	Управление экземплярами БД		Понимание модели OSI
Администрирование сети	Знания в области сетевых протоколов	Администрирование безопасности сети	Понимание стека протокола TCP/IP
	Знания в области маршрутизации		Протоколы шифрования и аутентификации
	Знания в области реализации		Антивирусные системы
			Планирование PKI

	VPN		
	Знания в системах биллинга		Системы контроля доступа
	Построение сетей		Инцидентный анализ
			Резервное копирование
Администрирование почтовых серверов	Знание Unix-систем	Администрирование систем интеграции и развертывания	Знание принципов непрерывной интеграций
	Знание ОС семейства Windows Server		Знание инструментов для интеграции
	Знание программ почтовых серверов		Знание систем контроля версий
	Модули проверки на вирусы и спам		Знание инструментов для развертывания
	Модули интеграции с базами данных		
	Знание программ клиентов электронной почты		
	Понимание стека протокола TCP/IP		
	Знания в области сетевых протоколов		

Таблица № 3

### Навыки работы с инструментами

Группа	Инструменты	Группа	Инструменты
IDE	Microsoft Visual Studio	Текстовые редакторы	Microsoft Visual Studio Code
	Eclipse		Sublime Text
	IntelliJ IDEA		Notepad++
	PhpStorm		Coda 2
	Xcode		Atom
	PyCharm		Brackets
	Dev-C++		Light Table
	WebStorm		Vim
	Komodo IDE		Dart Editor
	RubyMine		Системы контроля версий
	C++Builder	SVN	
	AndroidStudio	Mercurial	
	CodeLite	CVS	
	Code::Blocks	Team Foundation Server	
	NetBeans	Профилерование	Bazaar
	AptanaStudio		YourKit
	ShiftEdit		Jprofiler
	Cloud 9 IDE		xDebug
	MonoDevelop		XHProf
	SharpDevelop		Gperf
DrJava	hotspot		

	Xamarin Studio		Python profile
Интеграция и развертывание	Travis CI		dowser
	CircleCI		YUI profiler
	Railsonfile		
	Wercker		
	Hostedci		
	Jenkins		

Таблица № 4

Обобщенные и социальные навыки

Область	Навыки	Область	Навыки	
Структуры данных	Обычные	Алгоритмы	Компьютерная графика	
	Параллельные		Цифровая обработка сигналов	
	Императивные		Информационная безопасность	
	Функциональные		Анализ данных и машинное обучение	
Прикладной опыт	Сфера		Опыт работы с платформами	Искусственный интеллект
	Пользовательский интерфейс веб-систем (front-end)			Компьютерные сети
	Серверный код веб-систем (back-end)			Моделирование
	Разработка игр			Высокопроизводительные вычисления
	Разработка под мобильные платформы			Компьютерная математика
	Разработка для ПК			Распределенные информационные системы
	Высокопроизводительные вычисления	Microsoft .Net Framework		
	Администрирование баз данных	Android		
	Администрирование веб-сервера	iOS		
	Администрирование сети	Windows Phone		
	Администрирование безопасности сети	Blackberry		
	Администрирование почтовых серверов	Java (JDK and JRE)		
	Управление проектами	MAC		
	Тестирование	Unix		
Тестирование	Компонентное тестирование	Личные и	Silverlite	
	Модульное тестирование		Flash AIR	
	Интеграционное тестирование		Mono	
	Системное тестирование		LiveCode	
	Приемочное тестирование		Qt	
	Тестирование безопасности		Open Web Platform	
	Тестирование взаимодействия		Умение работать в группе	
	Функциональное тестирование		Инновационный/творческий ум	

	Нагрузочное тестирование	социальные навыки	Открытость и адаптивность к изменениям
	Тестирование стабильности и надежности		Критическое мышление
	Объемное тестирование		Анализ и решение проблем
	Тестирование установки		Планирование и контроль
	Тестирование удобства пользования		Умение слушать
	Тестирование на отказ и восстановление		Умение работать под давлением
	Конфигурационное тестирование		Надежность
Управление проектами	Управление конфликтами		Навык писателя
	Организация совещаний		Гибкость и адаптивность
	Управление временем		Способность эффективно задавать вопросы
	Построение команды		Управление временем
	Управление ожиданиями		Умение работать под давлением
	Гибкость и адаптивность		Надежность
	Анализ и решение проблем		

Данные таблицы навыков не являются окончательными. Система предполагает возможность ее расширения в соответствии со спецификой решаемых задач. Представление информации об обладании каким-либо навыком в виде функции принадлежности наилучшим образом отражает нечеткость данной характеристики. Поэтому использование теории нечетких множеств наиболее удобно при распределении людских ресурсов по задачам, исходя из имеющихся у разработчиков навыков и специфики задач.

Работа выполнена при поддержке РФФИ проект № 16-01-00390.

### Литература

1. Hastie S., Wojewoda S. Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch. Info, 2015. URL: [infoq.com/articles/standish-chaos-2015](http://infoq.com/articles/standish-chaos-2015).
2. Zhivaeva A.A. et al. Decision support systems for planning the software development process // Modern informatization problems in economics and safety / ed. Kravets O.J. et al. Yelm, WA, USA: Scince Boor Publishing House, 2017. pp. 101–105.

3. А.П. Ротштейн. Интеллектуальные технологии идентификации // Проектирование систем управления\Fuzzy Logic Toolbox. URL: [matlab.exponenta.ru/fuzzylogic/book5/1\\_1.php](http://matlab.exponenta.ru/fuzzylogic/book5/1_1.php).
4. Кофман А. Введение в теорию нечетких множеств / ред. С.И. Травкин. М.: Радио и связь, 1982. 432 с.
5. Гинис Л.А. Развитие инструментария когнитивного моделирования для исследования сложных систем // Инженерный вестник Дона. 2013. № 3. URL: [ivdon.ru/ru/magazine/archive/n3y2013/1806](http://ivdon.ru/ru/magazine/archive/n3y2013/1806).
6. Венцов Н.Н., Долгов В.В., Подколзина Л.А. Об одном способе построения запросов к базе данных на основе аппарата нечеткой логики // Инженерный вестник Дона. 2015. № 3. URL: [ivdon.ru/ru/magazine/archive/n3y2015/3172](http://ivdon.ru/ru/magazine/archive/n3y2015/3172).
7. Wen F., Lin C.-M. Multistage Human Resource Allocation for Software Development by Multiobjective Genetic Algorithm // Open Appl. Math. J. 2008. № 2. pp. 95.
8. Alison Doyle. Top Skills to List on LinkedIn. Most Popular Skills Employers Seek on LinkedIn. 2017. URL: [thebalance.com/top-skills-to-list-on-linkedin-2062321](http://thebalance.com/top-skills-to-list-on-linkedin-2062321).
9. Mtsweni E.S., Hörne T., Poll J.A. Van Der. Soft Skills for Software Project Team Members // Int. J. Comput. Theory Eng. 2016. Vol. 8, № 2. pp. 150–155.
10. Ahmed F. et al. What Soft Skills Software Architect Should Have? A Reflection from Software Industry // Int. Conf. Comput. Commun. Manag. 2011. Vol. 5. pp. 565–569.
11. Silva L.C. e, Costa A.P.C.S. Decision model for allocating human resources in information system projects // Int. J. Proj. Manag. 2013. Vol. 31, № 1. pp. 100–108.
12. Avinash A., Ramani K. A Hybrid Technique for Software Project Scheduling and Human Resource Allocation // Int. J. Eng. Dev. Res. 2014. Vol. 2, № 3. pp. 3243–3251.

## References

1. Hastie S., Wojewoda S. Standish Group 2015 Chaos Report. Q&A with Jennifer Lynch. Info, 2015. URL: [infoq.com/articles/standish-chaos-2015](http://infoq.com/articles/standish-chaos-2015).
2. Zhivaeva A.A. et al. Decision support systems for planning the software development process. Modern informatization problems in economics and safety. ed. Kravets O.J. et al. Yelm, WA, USA: Scince Boor Publishing House, 2017. pp. 101-105.
3. A.P. Rotshtejn. Intellektual'nye tehnologii identifikacii [Intelligent identification technologies]. Proektirovanie sistem upravlenija.Fuzzy Logic Toolbox. URL: [matlab.exponenta.ru/fuzzylogic/book5/1\\_1.php](http://matlab.exponenta.ru/fuzzylogic/book5/1_1.php).
4. Kofman A. Vvedenie v teoriju nechetkih mnozhestv [Introduction to the theory of fuzzy sets]. ed. S.I. Travkin. M.: Radio i svjaz', 1982. 432 p.
5. Ginis L.A. Inženernyj vestnik Dona (Rus), 2013. № 3. URL: [ivdon.ru/ru/magazine/archive/n3y2013/1806](http://ivdon.ru/ru/magazine/archive/n3y2013/1806).
6. Vencov N.N., Dolgov V.V., Podkolzina L.A. Inženernyj vestnik Dona (Rus), 2015. № 3. URL: [ivdon.ru/ru/magazine/archive/n3y2015/3172](http://ivdon.ru/ru/magazine/archive/n3y2015/3172).
7. Wen F., Lin C.-M. Multistage Human Resource Allocation for Software Development by Multiobjective Genetic Algorithm. Open Appl. Math. J. 2008. № 2. pp. 95.
8. Alison Doyle. Top Skills to List on LinkedIn. Most Popular Skills Employers Seek on LinkedIn. 2017. URL: [thebalance.com/top-skills-to-list-on-linkedin-2062321](http://thebalance.com/top-skills-to-list-on-linkedin-2062321).
9. Mtsweni E.S., Hörne T., Poll J.A. Van Der. Soft Skills for Software Project Team Members. Int. J. Comput. Theory Eng. 2016. Vol. 8, № 2. pp. 150-155.
10. Ahmed F. et al. What Soft Skills Software Architect Should Have? A Reflection from Software Industry. Int. Conf. Comput. Commun. Manag. 2011. Vol. 5. pp. 565-569.
11. Silva L.C. e, Costa A.P.C.S. Decision model for allocating human resources in information system projects. Int. J. Proj. Manag. 2013. Vol. 31, № 1.



pp. 100-108.

12. Avinash A., Ramani K. A Hybrid Technique for Software Project Scheduling and Human Resource Allocation. Int. J. Eng. Dev. Res. 2014. Vol. 2, № 3. pp. 3243-3251.