

Глава 11: Профессиональная практика разработки программного обеспечения

Из SWEBOK

Содержание

- 1 Профессионализм
 - 1.1 Аккредитация, сертификация и лицензирование
 - 1.2 Кодексы этики и профессионального поведения
 - 1.3 Природа и роль профессиональных обществ
 - 1.4 Природа и роль стандартов программной инженерии
 - 1.5 Экономический эффект программного обеспечения
 - 1.6 Трудовые договоры
 - 1.7 Правовые вопросы
 - 1.8 Документация
 - 1.9 Анализ компромиссов
- 2 Групповая динамика и психология
 - 2.1 Динамика работы в командах/группах
 - 2.2 Индивидуальное познание
 - 2.3 Работа со сложностью проблемы
 - 2.4 Взаимодействие с заинтересованными сторонами
 - 2.5 Работа с неопределенностью и двусмысленностью
 - 2.6 Работа в мультикультурной среде
- 3 навыка общения
 - 3.1 Чтение, понимание и обобщение
 - 3.2 Написание
 - 3.3 Командное и групповое общение
 - 3.4 Навыки презентации

АКРОНИМЫ

| | |
|-----------------------------------|---|
| АКМ | Ассоциация вычислительной техники |
| БКС | Британское компьютерное общество |
| CSDA | Сертифицированный специалист по разработке программного обеспечения |
| ОПБО | Сертифицированный специалист по разработке программного обеспечения |
| МЭК | Международная электротехническая комиссия |
| IEEE CS | Компьютерное общество IEEE |
| ИФИП | Международная федерация обработки информации |
| IP | Интеллектуальная собственность |
| ИСО | Международная организация по стандартизации |
| Соглашение о неразглашении | Соглашение о неразглашении |
| о неразглашении | Всемирная организация интеллектуальной собственности |
| ВОИС | |
| ВТО | Мировая Торговая Организация |

ВВЕДЕНИЕ

Область знаний «Профессиональная практика разработки программного обеспечения» (КА) связана со знаниями, навыками и отношением, которыми инженеры-программисты должны обладать, чтобы заниматься разработкой программного обеспечения профессионально, ответственно и этично. Из-за широкого применения программных продуктов в общественной и личной жизни качество программных продуктов может иметь огромное влияние на наше личное благополучие и социальную гармонию. Инженеры-программисты должны решать уникальные инженерные задачи, создавая программное обеспечение с известными характеристиками и надежностью. Это требование требует от инженеров-программистов надлежащего набора знаний, навыков, подготовки и опыта профессиональной деятельности.

Термин «профессиональная практика» относится к способу оказания услуг с целью достижения определенных стандартов или критериев как в процессе оказания услуги, так и в конечном продукте, получаемом в результате услуги. Эти стандарты и критерии могут включать как технические, так и нетехнические аспекты. Концепцию профессиональной практики можно рассматривать как более применимую к тем профессиям, которые обладают общепринятой совокупностью знаний; кодексы этики и профессионального поведения с наказаниями за нарушения; принятые процессы аккредитации, сертификации и лицензирования; и профессиональные общества для обеспечения и управления всем этим. Прием в эти профессиональные общества часто зависит от предписанного сочетания образования и опыта.

Инженер-программист поддерживает профессиональную практику, выполняя всю работу в соответствии с общепринятыми практиками, стандартами и рекомендациями, в частности, установленными соответствующим профессиональным сообществом. Например, Ассоциация вычислительной техники (ACM) и Компьютерное общество IEEE (IEEE CS) разработали Кодекс этики и профессиональной практики разработки программного обеспечения. И Британское компьютерное общество (BCS), и Международная федерация обработки информации (IFIP) установили аналогичные стандарты профессиональной практики. ISO/IEC и IEEE также предоставили международно признанные стандарты разработки программного обеспечения (см. Приложение В к этому *Руководству*). IEEE CS учредила две международные программы сертификации (CSDA, CSDP) и соответствующую *Руководство по своду знаний по программной инженерии (Руководство SWEBOOK)*. Все это элементы, которые закладывают основу профессиональной практики разработки программного обеспечения.

РАЗБИВКА ТЕМ ДЛЯ ПРОФЕССИОНАЛЬНОЙ ПРАКТИКИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Разбивка по темам профессиональной практики разработки программного обеспечения КА показана на рис. 11.1. Подобластями, представленными в этом КА, являются профессионализм, групповая динамика и психология, а также коммуникативные навыки.

1 Профессионализм

Инженер-программист демонстрирует профессионализм, в частности, благодаря соблюдению кодексов этики и профессионального поведения, а также стандартов и практики, установленных профессиональным сообществом инженеров.

Профессиональное сообщество часто представлено одним или несколькими профессиональными обществами; эти общества публикуют кодексы этики и профессионального поведения, а также критерии приема в сообщество. Эти критерии составляют основу деятельности по аккредитации и лицензированию и могут использоваться в качестве меры для определения инженерной компетентности или небрежности.

1.1 Аккредитация, сертификация и лицензирование

[1 , ccl54.1, c1s5.1–c1s5.4]

1.1.1 Аккредитация

Аккредитация – это процесс подтверждения компетентности, авторитета или надежности организации. Аккредитованные школы или программы гарантированно придерживаются определенных стандартов и поддерживают определенные качества. Во многих странах основным средством, с помощью которого инженеры приобретают знания, является прохождение аккредитованного курса обучения. Часто инженерную аккредитацию проводит государственная организация, например министерство образования. К таким странам с государственной аккредитацией относятся Китай, Франция, Германия, Израиль, Италия и Россия.

Однако в других странах процесс аккредитации не зависит от правительства и осуществляется частными ассоциациями. Например, в Соединенных Штатах инженерная аккредитация осуществляется организацией, известной как ABET. Организация, известная как CSAB, выступающая в качестве органа-участника ABET, является ведущим обществом ABET по аккредитации программ на получение степени в области разработки программного обеспечения. Хотя процесс аккредитации может быть разным для каждой страны и юрисдикции, общий смысл одинаков. Аккредитация учебного заведения означает, что «орган по аккредитации признает образовательное учреждение поддерживающим стандарты, дающие право выпускникам на поступление в высшие или более специализированные учебные заведения или на профессиональную практику» [2].

1.1.2 Сертификация

Сертификация относится к подтверждению конкретных характеристик человека. Распространенным типом сертификации является профессиональная сертификация, когда лицо сертифицируется как способное выполнять деятельность в определенной дисциплине на установленном уровне компетенции. Профессиональная сертификация также может подтвердить способность владельца соответствовать

профессиональным стандартам и применять профессиональные суждения при решении или решении проблем. Профессиональная сертификация может также включать проверку предписанных знаний, освоение передовой практики и проверенных методологий, а также объем профессионального опыта.

Инженер обычно получает сертификат, сдав экзамен в сочетании с другими критериями, основанными на опыте. Эти экзамены часто проводятся неправительственными организациями, такими как профессиональные общества.

В программной инженерии сертификация свидетельствует о квалификации инженера-программиста. Например, IEEE CS ввела в действие две программы сертификации (CSDA и CSDP), предназначенные для подтверждения знаний инженера-программиста стандартных методов разработки программного обеспечения и продвижения по карьерной лестнице. Отсутствие сертификации не исключает возможности работы инженером-программистом. В настоящее время сертификация в области разработки программного обеспечения является полностью добровольной. На самом деле, большинство инженеров-программистов не сертифицировано ни по одной программе.



1.1.3 Лицензирование

«Лицензирование» — это действие по предоставлению лицу полномочий на выполнение определенных видов деятельности и принятие на себя ответственности за полученные инженерные продукты. Существительное «лицензия» относится как к этому разрешению, так и к документу, фиксирующему это разрешение. Правительственные органы или законодательные органы обычно выдают лицензии.

Для получения лицензии на практику требуется, чтобы человек не только соответствовал определенному стандарту, но и обладал определенными способностями к практике или работе. Иногда существует требование начального уровня, которое устанавливает минимальные навыки и способности для практики, но по мере того, как профессионал продвигается по своей карьере, требуемые навыки и способности меняются и развиваются.

Как правило, инженеры лицензируются как средство защиты населения от неквалифицированных лиц. В некоторых странах никто не может работать профессиональным инженером без лицензии; или, кроме того, ни одна компания не может предлагать «инженерные услуги», если там не работает по крайней мере один лицензированный инженер.

1.2 Кодексы этики и профессионального поведения

[1 , c1s6–c1s9] [3 , c8] [4 , c1s2] [5 , c33] [6]

Кодексы этики и профессионального поведения включают в себя ценности и поведение, которые должны воплощаться в профессиональной практике и решениях инженера.

Профессиональное сообщество устанавливает кодексы этики и профессионального поведения. Они существуют в контексте социальных норм и местных законов и согласовываются с ними. Таким образом, кодексы этики и профессионального поведения представляют собой руководство перед лицом противоречивых императивов.

После создания кодексы этики и профессионального поведения обеспечиваются соблюдением профессии, представленной профессиональными обществами или установленным законом органом.

Нарушениями могут быть акты совершения, такие как сокрытие ненадлежащей работы, раскрытие конфиденциальной информации, фальсификация информации или искажение своих способностей. Они также могут возникать из-за упущения, в том числе из-за нераскрытия рисков или предоставления важной информации, из-за ненадлежащего указания авторства или подтверждения рекомендаций, а также из-за непредставления интересов клиента. Нарушения кодексов этики и профессионального поведения могут повлечь за собой санкции и возможное отстранение от профессионального статуса.

Кодекс этики и профессионального поведения в области разработки программного обеспечения был одобрен Советом ACM и Советом управляющих IEEE CS в 1999 г. [6*]. Согласно короткой версии этого кода:

- Инженеры-программисты обязуются сделать анализ, спецификацию, проектирование, разработку, тестирование и обслуживание программного обеспечения полезной и уважаемой профессией. В соответствии со своей приверженностью здоровью, безопасности и благополучию населения разработчики программного обеспечения должны придерживаться восьми принципов, касающихся общественности, клиента и работодателя, продукта, суждения, управления, профессии, коллег и самого себя соответственно.

Поскольку стандарты и кодексы этики и профессионального поведения могут быть введены, изменены или заменены в любое время, отдельные инженеры-программисты несут ответственность за свое постоянное обучение, чтобы оставаться в курсе своей профессиональной деятельности.

1.3 Природа и роль профессиональных обществ

[1 , c1s1–c1s2] [4 , c1s2] [5 , c35s1]

Профессиональные общества состоят из практиков и ученых. Эти общества служат для определения, продвижения и регулирования соответствующих профессий. Профессиональные общества помогают устанавливать профессиональные стандарты, а также кодексы этики и профессионального поведения. По этой причине они также занимаются сопутствующими видами деятельности, в том числе

- установление и распространение свода общепринятых знаний;
- аккредитация, сертификация и лицензирование;
- наложение дисциплинарных взысканий;
- продвижение профессии посредством конференций, тренингов и публикаций.

Участие в профессиональных сообществах помогает отдельному инженеру поддерживать и оттачивать свои профессиональные знания и актуальность, а также расширять и поддерживать свою профессиональную сеть.

1.4 Природа и роль стандартов программной инженерии

[1 , c5s3.2, c10s2.1] [5 , c32s6] [7 , c1s2]

Стандарты разработки программного обеспечения охватывают удивительное разнообразие тем. Они предоставляют рекомендации по практике разработки программного обеспечения и процессы, которые будут использоваться во время разработки, обслуживания и поддержки программного обеспечения. Устанавливая согласованную совокупность знаний и опыта, стандарты разработки программного обеспечения создают основу, на которой могут разрабатываться дальнейшие рекомендации. Приложение В к этому Руководству *содержит рекомендации по стандартам* разработки программного обеспечения IEEE и ISO/IEC, которые поддерживают области знаний, изложенные в этом *Руководстве*.

Преимущества стандартов разработки программного обеспечения многочисленны и включают улучшение качества программного обеспечения, помощь в предотвращении ошибок, защиту как производителей программного обеспечения, так и пользователей, повышение профессиональной дисциплины и помощь в переходе на новые технологии.

1.5 Экономический эффект программного обеспечения

[3 , c10s8] [4 , c1s1.1] [8 , c1]

Программное обеспечение имеет экономический эффект на индивидуальном, деловом и общественном уровнях. «Успех» программного обеспечения может определяться пригодностью продукта для решения выявленной проблемы, а также его эффективностью применительно к этой проблеме.

На индивидуальном уровне продолжение работы инженера может зависеть от его способности и готовности интерпретировать и выполнять задачи, отвечающие потребностям и ожиданиям клиентов или работодателей. В свою очередь, покупка программного обеспечения может положительно или отрицательно повлиять на финансовое положение клиента или работодателя.

На уровне бизнеса программное обеспечение, правильно примененное к проблеме, может сократить месяцы работы и привести к увеличению прибыли или повышению эффективности организации. Более того, организации, которые приобретают или предоставляют успешное программное обеспечение, могут быть благом для общества, в котором они работают, предоставляя как рабочие места, так и более качественные услуги. Однако затраты на разработку или приобретение программного обеспечения также могут быть равны затратам на любое крупное приобретение.

На общественном уровне прямое воздействие успеха или неудачи программного обеспечения включает или исключает несчастные случаи, перебои и потерю обслуживания. Косвенные воздействия включают успех или неудачу организации, которая приобрела или произвела программное обеспечение, повышение или снижение общественной производительности, гармоничный или разрушительный общественный порядок и даже спасение или потерю имущества и жизни.

1.6 Трудовые договоры

[1 , c7]

Услуги по разработке программного обеспечения могут предоставляться в рамках различных отношений между клиентом и разработчиком. Работа по разработке программного обеспечения может быть предложена в качестве поставщика от компании к заказчику, консультирования между инженером и заказчиком, прямого найма или даже волонтерства. Во всех этих ситуациях заказчик и поставщик договариваются о том, что продукт или услуга будут предоставлены в обмен на определенное вознаграждение. Здесь нас больше всего интересует договоренность между инженером и заказчиком и сопутствующие соглашения или контракты, независимо от того, относятся ли они к прямому найму или консультанту, а также вопросы, которые они обычно решают.

Общей проблемой в контрактах на разработку программного обеспечения является конфиденциальность. Работодатели получают коммерческую выгоду от интеллектуальной собственности, поэтому они стремятся защитить эту собственность от раскрытия. Поэтому от разработчиков программного обеспечения часто требуется подписать соглашение о неразглашении (NDA) или об интеллектуальной собственности (IP) в качестве предварительного условия для работы. Эти соглашения обычно применяются к информации, которую инженер-программист может получить только благодаря сотрудничеству с заказчиком. Срок действия этих соглашений может распространяться после прекращения ассоциации.

Другая проблема связана с владением ИС. Права на активы разработки программного обеспечения — продукты, инновации, изобретения, открытия и идеи — могут принадлежать работодателю или заказчику либо в соответствии с явными условиями контракта, либо в соответствии с применимыми законами, если эти активы получены в течение срока действия отношений инженера-программиста с этим работодателем. или клиент. Контракты различаются по праву собственности на активы, созданные с использованием оборудования или информации, не принадлежащих работодателю.

Наконец, в контрактах может также указываться, помимо прочего, место, где должна выполняться работа; стандарты, по которым будет проводиться эта работа; конфигурация системы, которая будет использоваться для разработки; ограничение ответственности инженера-программиста и работодателя; коммуникационная матрица и/или план эскалации; и административные детали, такие как ставки, частота вознаграждения, рабочее время и условия труда.

1.7 Правовые вопросы

[1 , c6, c11] [3 , c5s3–c5s4] [9 , c1s10]

Правовые вопросы, связанные с профессиональной практикой разработки программного обеспечения, в частности, включают вопросы, связанные со стандартами, товарными знаками, патентами, авторскими правами, коммерческой тайной, профессиональной ответственностью, юридическими требованиями, соблюдением правил торговли и киберпреступностью. Поэтому полезно обладать знаниями по этим вопросам и их применимости.

Юридические вопросы основаны на юрисдикции; инженеры-программисты должны консультироваться с адвокатами, которые специализируются на типе и юрисдикции любых выявленных юридических вопросов.

1.7.1 Стандарты

Стандарты разработки программного обеспечения устанавливают руководящие принципы для общепринятых практик и минимальных требований к продуктам и услугам, предоставляемым инженером-программистом. Приложение В этого *руководства* содержит рекомендации по стандартам разработки программного обеспечения, которые применимы к каждому КА.

Стандарты являются ценным источником требований и помощи в повседневной деятельности по разработке программного обеспечения. Соблюдение стандартов способствует дисциплине за счет перечисления минимальных характеристик продуктов и практики. Эта дисциплина помогает смягчить подсознательные предположения или чрезмерную уверенность в дизайне. По этим причинам организации, занимающиеся разработкой программного обеспечения, часто включают соответствие стандартам в свою организационную политику. Кроме того, соблюдение стандартов является основным компонентом защиты от судебных исков или обвинений в злоупотреблении служебным положением.

1.7.2 Товарные знаки

Товарный знак относится к любому слову, имени, символу или устройству, которое используется в деловых операциях. Он используется «для указания источника или происхождения товара» [2].

Защита товарных знаков защищает имена, логотипы, изображения и упаковку. Однако, если имя, изображение или другой товарный знак становится общим термином, охрана товарного знака аннулируется.

Всемирная организация интеллектуальной собственности (ВОИС) является органом, который устанавливает правила и нормы в отношении товарных знаков. ВОИС — это учреждение Организации Объединенных Наций, занимающееся использованием интеллектуальной собственности как средства стимулирования инноваций и творчества.

1.7.3 Патенты

Патенты защищают право изобретателя производить и продавать идеи. Патент состоит из набора исключительных прав, предоставленных суверенным правительством отдельному лицу, группе лиц или организации на ограниченный период времени. Патенты — это старая форма защиты прав собственности на идеи, которая восходит к 15 веку.

Подача заявки на патент требует тщательного учета процесса, приведшего к изобретению. Патентные поверенные помогают в написании заявлений о раскрытии патента таким образом, который с наибольшей вероятностью защитит права инженера-программиста.

Обратите внимание, что если изобретения сделаны в ходе контракта на разработку программного обеспечения, право собственности может принадлежать работодателю или заказчику или находиться в совместном владении, а не принадлежать инженеру-программисту.

Существуют правила относительно того, что является патентоспособным, а что нет. Во многих странах программный код не подлежит патентованию, хотя программные алгоритмы могут быть патентоспособными. Существующие и поданные патентные заявки можно найти в ВОИС.

1.7.4 Авторские права

Большинство правительств в мире предоставляют исключительные права на оригинальное произведение его создателю, как правило, на ограниченный срок, что закреплено в качестве авторского права. Авторские права защищают способ представления идеи, а не саму идею. Например, они могут охранять конкретную формулировку отчета об историческом событии, тогда как само событие не защищено. Авторские права являются долгосрочными и возобновляемыми; они датируются 17 веком.

1.7.5 Коммерческая тайна

Во многих странах такие интеллектуальные активы, как формулы, алгоритмы, процессы, конструкции, методы, шаблоны, инструменты или подборки информации, могут считаться «коммерческой тайной» при условии, что эти активы не являются общеизвестными и могут предоставлять коммерческую тайну. некоторое экономическое преимущество. Обозначение «коммерческая тайна» обеспечивает правовую защиту в случае кражи актива. Эта защита не ограничена во времени. Однако, если другая сторона получит или обнаружит тот же актив на законных основаниях, то этот актив больше не будет защищен, и другая сторона также будет обладать всеми правами на его использование.

1.7.6 Профессиональная ответственность

Инженеры-программисты обычно озабочены вопросами профессиональной ответственности. Поскольку физическое лицо предоставляет услуги клиенту или работодателю, жизненно важно придерживаться стандартов и общепринятых практик, тем самым защищая от обвинений или судебных разбирательств в связи с злоупотреблением служебным положением, небрежностью или некомпетентностью.

Для инженеров, включая инженеров-программистов, профессиональная ответственность связана с ответственностью за продукт. В соответствии с законами и правилами, действующими в их юрисдикции, инженеры могут быть привлечены к ответственности за неполное и добросовестное соблюдение рекомендуемой практики; это известно как «халатность». На них также могут распространяться законы, регулирующие «строгую ответственность» и либо подразумеваемые, либо явно выраженные гарантии, когда, продавая продукт, инженер должен гарантировать, что продукт подходит и безопасен для использования. В некоторых странах (например, в США) «конфиденциальность» (идея о том, что подать в суд можно только на лицо, продающее продукт) больше не является защитой от исков об ответственности.

Судебные иски об ответственности могут быть поданы в соответствии с деликтным правом в США, что позволяет любому, кто пострадал, возместить свои убытки, даже если не было предоставлено никаких гарантий. Поскольку трудно измерить пригодность или безопасность программного обеспечения, несоблюдение должной осторожности может быть использовано для доказательства небрежности со стороны разработчиков программного обеспечения. Защита от такого обвинения заключается в том, чтобы показать, что при разработке продукта соблюдались стандарты и общепринятые методы.

1.7.7 Юридические требования

Инженеры-программисты должны действовать в рамках местных, национальных и международных правовых норм. Поэтому инженеры-программисты должны быть осведомлены о юридических требованиях к

- регистрация и лицензирование, включая экзамены, образование, опыт и требования к обучению;

- договорные соглашения;
- внедоговорные правовые нормы, такие как те, которые регулируют ответственность;
- Базовую информацию о международно-правовой базе можно получить во Всемирной торговой организации (ВТО).

1.7.8 Торговое соответствие

Все специалисты по программному обеспечению должны знать о юридических ограничениях на импорт, экспорт или реэкспорт товаров, услуг и технологий в юрисдикциях, в которых они работают. Соображения включают экспортный контроль и классификацию, передачу товаров, приобретение необходимых государственных лицензий на иностранное использование аппаратного и программного обеспечения, услуг и технологий государством, предприятием или отдельными лицами, находящимися под санкциями, а также импортные ограничения и пошлины. Для получения подробного руководства по соблюдению требований следует проконсультироваться с торговыми экспертами.

1.7.9 Киберпреступность

Киберпреступность относится к любому преступлению, связанному с компьютером, компьютерным программным обеспечением, компьютерными сетями или встроенным программным обеспечением, управляющим системой. Компьютер или программное обеспечение могли быть использованы при совершении преступления или могли быть целью. Эта категория преступлений включает мошенничество, несанкционированный доступ, спам, непристойный или оскорбительный контент, угрозы, домогательства, кражу конфиденциальных личных данных или коммерческой тайны, а также использование одного компьютера для повреждения или проникновения в другие сетевые компьютеры и автоматизированные средства управления системой.

Пользователи компьютеров и программного обеспечения совершают мошенничество, изменяя электронные данные для облегчения незаконной деятельности. Формы несанкционированного доступа включают взлом, прослушивание и использование компьютерных систем способом, скрытым от их владельцев.

Во многих странах существуют отдельные законы о киберпреступлениях, но иногда бывает сложно преследовать киберпреступления в судебном порядке из-за отсутствия четко сформулированных законов. Инженер-программист обязан учитывать угрозу киберпреступности и понимать, как программная система будет защищать или подвергать опасности программное обеспечение и пользовательскую информацию от случайного или злонамеренного доступа, использования, модификации, уничтожения или раскрытия.

1.8 Документация

[1 , с10s5.8] [3 , с1s5] [5 , с32]

Предоставление четкой, тщательной и точной документации является обязанностью каждого инженера-программиста. Адекватность документации оценивается по различным критериям, основанным на потребностях различных аудиторий заинтересованных сторон.

Хорошая документация соответствует принятым стандартам и руководствам. В частности, инженеры-программисты должны документировать

- соответствующие факты,
- значительные риски и компромиссы, а также
- предупреждения о нежелательных или опасных последствиях использования или неправильного использования программного обеспечения.

Инженеры-программисты должны избегать

- сертификация или одобрение неприемлемых продуктов,
- раскрытие конфиденциальной информации или
- фальсификация фактов или данных.

Кроме того, инженеры-программисты и их менеджеры должны, в частности, предоставить следующую документацию для использования другими подразделениями организации по разработке программного обеспечения:

- спецификации требований к программному обеспечению, документы по проектированию программного обеспечения, подробные сведения об используемых инструментах разработки программного обеспечения, спецификации и результаты тестирования программного обеспечения, а также сведения о принятых методах разработки программного обеспечения;
- проблемы, возникшие в процессе разработки.

Для внешних заинтересованных сторон (клиентов, пользователей и др.) документация по программному обеспечению должна, в частности, обеспечивать

- информация, необходимая для определения того, может ли программное обеспечение удовлетворить потребности клиентов и пользователей,
- описание безопасного и небезопасного использования программного обеспечения,
- описание защиты конфиденциальной информации, созданной или хранимой с помощью программного обеспечения, и
- четкая идентификация предупреждений и критических процедур.

Использование программного обеспечения может включать установку, эксплуатацию, администрирование и выполнение других функций различными группами пользователей и вспомогательного персонала. Если заказчик приобретет право собственности на исходный код программного обеспечения или право на изменение кода, инженер-программист должен предоставить документацию по функциональным спецификациям, дизайну программного обеспечения, набору тестов и необходимой операционной среде для программного обеспечения.

Минимальная продолжительность хранения документов составляет продолжительность жизненного цикла программных продуктов или время, требуемое соответствующими организационными или нормативными требованиями.

1.9 Анализ компромиссов

[3 , c1c2, c10] [9 , c9c5.10]

В практике разработки программного обеспечения инженеру-программисту часто приходится выбирать между альтернативными решениями проблемы. Результат этого выбора определяется профессиональной оценкой разработчиком программного обеспечения рисков, затрат и преимуществ альтернатив в сотрудничестве с заинтересованными сторонами. Оценка инженера-программиста называется «анализом компромиссов». Анализ компромиссов, в частности, позволяет идентифицировать конкурирующие и взаимодополняющие требования к программному обеспечению, чтобы определить приоритет окончательного набора требований, определяющих программное обеспечение, которое будет создано (см. Согласование требований в Требованиях к программному обеспечению КА и Определение и согласование требований в Управлении программной инженерией КА).

В случае текущего проекта разработки программного обеспечения, который опаздывает или выходит за рамки бюджета, часто проводится анализ компромиссов, чтобы решить, какие требования к программному обеспечению могут быть смягчены или исключены с учетом их последствий.

Первым шагом в анализе компромиссов является установление целей проектирования (см. «Инженерный дизайн» в «Основах проектирования») и определение относительной важности этих целей. Это позволяет определить решение, наиболее точно соответствующее этим целям; это означает, что способ формулирования целей имеет решающее значение.

Цели проектирования могут включать в себя минимизацию денежных затрат и максимизацию надежности, производительности или некоторых других критериев в широком диапазоне измерений. Однако трудно сформулировать анализ компромисса затрат и риска, особенно когда первичные производственные и вторичные затраты, основанные на риске, должны быть сопоставлены друг с другом.

Инженер-программист должен проводить анализ компромиссов с соблюдением этических норм, в частности, будучи объективным и беспристрастным при выборе критериев для сравнения альтернативных решений проблем и при присвоении весов или важности этим критериям. Любой конфликт интересов должен быть раскрыт заранее.

2 Групповая динамика и психология

Инженерная работа очень часто ведется в контексте командной работы. Инженер-программист должен уметь сотрудничать и конструктивно взаимодействовать с другими, чтобы сначала определить, а затем удовлетворить как потребности, так и ожидания. Знание групповой динамики и психологии является преимуществом при взаимодействии с клиентами, коллегами, поставщиками и подчиненными для решения проблем разработки программного обеспечения.

2.1 Динамика работы в командах/группах

[3 , c1c6] [9 , c1c3.5, c10]

Инженеры-программисты должны работать вместе с другими. С одной стороны, они работают внутри инженерных групп; с другой стороны, они работают с клиентами, представителями общественности, регулирующими органами и другими заинтересованными сторонами. Эффективные команды — те, которые демонстрируют постоянное качество работы и прогресс в достижении целей, — сплочены и обладают совместной, честной и целенаправленной атмосферой. Индивидуальные и командные цели согласованы таким образом, что участники естественным образом принимают на себя обязательства и чувствуют свою ответственность за общие результаты.

Члены команды способствуют созданию этой атмосферы, будучи интеллектуально честными, используя групповое мышление, признавая свое невежество и признавая ошибки. Они справедливо распределяют ответственность, вознаграждения и рабочую нагрузку. Они заботятся о том, чтобы общаться четко, напрямую друг с другом и в документах, а также в исходном коде, чтобы информация была доступна для всех. Экспертные отзывы о рабочих продуктах оформлены конструктивно и нелично (см. Обзоры и аудиты в КА качества программного обеспечения). Это позволяет всем членам проводить цикл непрерывного совершенствования и роста без личного риска. В целом члены сплоченных команд демонстрируют уважение друг к другу и своему лидеру.

Следует подчеркнуть, что инженеры-программисты должны уметь работать в мультидисциплинарных средах и в различных предметных областях. Поскольку сегодня программное обеспечение повсюду, от телефона до автомобиля, оно влияет на жизнь людей далеко за пределы более традиционной концепции программного обеспечения, предназначенного для управления информацией в бизнес-среде.

2.2 Индивидуальное познание

[3 , c1c6.5] [5 , c33]

Инженеры хотят решать проблемы. Способность решать проблемы эффективно и результативно — это то, к чему стремится каждый инженер. Однако ограничения и процессы индивидуального познания влияют на решение проблем. В разработке программного обеспечения, в частности из-за весьма абстрактной природы самого программного обеспечения, индивидуальное познание играет очень заметную роль в решении проблем.

В целом способность человека (в частности, инженера-программиста) разложить проблему на части и творчески разработать решение может быть ограничена

- потребность в дополнительных знаниях,
- подсознательные предположения,
- объем данных,
- страх неудачи или последствия неудачи,
- культура, будь то домен приложения или организация,
- отсутствие возможности выразить проблему,
- ощущаемая рабочая атмосфера и
- эмоциональное состояние личности.

2.3 Работа со сложностью проблемы

Г 2 02021 Г 5 0221

[3 , c3c2] [3 , c33]

Многие, если не большинство, проблем программной инженерии слишком сложны и трудны для решения в целом или для отдельных инженеров-программистов. Когда возникают такие обстоятельства, обычным средством принятия является командная работа и декомпозиция проблемы (см. Методы решения проблем в Computing Foundations KA).

Команды работают вместе, чтобы решать сложные и крупные проблемы, разделяя бремя и опираясь на знания и творческий потенциал друг друга. Когда инженеры-программисты работают в командах, разные взгляды и способности отдельных инженеров дополняют друг друга и помогают создать решение, которое иначе было бы трудно найти. Некоторыми конкретными примерами командной работы в разработке программного обеспечения являются парное программирование (см. Agile-методы в KA Модели и методы разработки программного обеспечения) и проверка кода (см. Обзоры и аудиты в KA качества программного обеспечения).

2.4 Взаимодействие с заинтересованными сторонами

[9 , c2s3.1]

Успех разработки программного обеспечения зависит от позитивного взаимодействия с заинтересованными сторонами. Они должны обеспечивать поддержку, информацию и обратную связь на всех этапах процесса жизненного цикла программного обеспечения. Например, на ранних стадиях очень важно определить всех заинтересованных лиц и узнать, как продукт повлияет на них, чтобы можно было правильно и полностью зафиксировать достаточное определение требований заинтересованных сторон.

Во время разработки заинтересованные стороны могут предоставлять отзывы о спецификациях и/или ранних версиях программного обеспечения, изменении приоритета, а также уточнении подробных или новых требований к программному обеспечению. Наконец, во время обслуживания программного обеспечения и до конца срока службы продукта заинтересованные стороны предоставляют отзывы об изменяющихся или новых требованиях, а также отчеты о проблемах, чтобы программное обеспечение могло быть расширено и улучшено.

Поэтому жизненно важно поддерживать открытое и продуктивное общение с заинтересованными сторонами на протяжении всего жизненного цикла программного продукта.

2.5 Работа с неопределенностью и двусмысленностью

[3 , c3c2] [5 , c33]

Как и инженерам в других областях, инженерам-программистам часто приходится иметь дело с неопределенностями и неясностями и разрешать их при предоставлении услуг и разработке продуктов. Инженер-программист должен атаковать и уменьшать или устранять любые неясности, которые мешают выполнению работы. Часто неопределенность является просто отражением недостатка знаний. В этом случае расследование с использованием официальных источников, таких как учебники и профессиональные журналы, интервью с заинтересованными сторонами или консультации с товарищами по команде и коллегами, может преодолеть это.

Когда неопределенность или двусмысленность не могут быть легко преодолены, разработчики программного обеспечения или организации могут рассматривать это как проектный риск. В этом случае оценки работ или цены корректируются, чтобы снизить ожидаемые затраты на их решение (см. Управление рисками в KA Управления программной инженерии).

2.6 Работа в мультикультурной среде

[9 , c10c7]

Мультикультурная среда может оказывать влияние на динамику группы. Это особенно верно, когда группа географически разделена или общение происходит нечасто, поскольку такое разделение повышает важность каждого контакта. Межкультурная коммуникация становится еще более сложной, если разница в часовых поясах делает устное общение менее частым.

Мультикультурная среда довольно распространена в разработке программного обеспечения, возможно, больше, чем в других областях техники, из-за сильной тенденции международного аутсорсинга и легкой мгновенной доставки программных компонентов по всему миру. Например, довольно часто

программный проект делится на части, невзирая на национальные и культурные границы, и также довольно часто команда программного проекта состоит из людей из разных культур.

Чтобы программный проект был успешным, члены команды должны достичь определенного уровня терпимости, признавая, что некоторые правила зависят от социальных норм и что не все общества получают одинаковые решения и ожидания.

Этой терпимости и сопутствующему пониманию может способствовать поддержка руководства и руководства. Более частое общение, в том числе личные встречи, может помочь смягчить географические и культурные различия, способствовать сплоченности и повысить производительность. Кроме того, возможность общаться с товарищами по команде на их родном языке может быть очень полезной.

3 навыка общения

Очень важно, чтобы инженер-программист хорошо общался как устно, так и в чтении и письме. Успешное выполнение требований к программному обеспечению и сроков зависит от четкого понимания между инженером-программистом и заказчиками, руководителями, коллегами и поставщиками. Оптимальное решение проблем становится возможным благодаря способности исследовать, понимать и обобщать информацию. Приемка продукта клиентом и безопасное использование продукта зависят от предоставления соответствующего обучения и документации. Из этого следует, что на собственный карьерный успех инженера-программиста влияет способность последовательно и эффективно и своевременно обеспечивать устную и письменную коммуникацию.

3.1 Чтение, понимание и обобщение

[5 , с33]

Инженеры-программисты умеют читать и понимать технические материалы. Технические материалы включают справочники, руководства, исследовательские работы и исходный код программы.

Чтение — это не только основной способ улучшения навыков, но и способ сбора информации, необходимой для выполнения инженерных задач. Инженер-программист просеивает накопленную информацию, отфильтровывая части, которые будут наиболее полезными. Клиенты могут потребовать, чтобы инженер-программист обобщил для них результаты сбора такой информации, упростив или объяснив ее, чтобы они могли сделать окончательный выбор между конкурирующими решениями.

Чтение и понимание исходного кода также является компонентом сбора информации и решения проблем. При изменении, расширении или переписывании программного обеспечения очень важно понимать как его реализацию, непосредственно полученную из представленного кода, так и его структуру, о которой часто приходится делать выводы.

3.2 Написание

[3 , с1с5]

Инженеры-программисты могут создавать письменные продукты в соответствии с требованиями клиентов или общепринятой практикой. Эти письменные продукты могут включать исходный код, планы проектов программного обеспечения, документы с требованиями к программному обеспечению, анализы рисков, документы по проектированию программного обеспечения, планы тестирования программного обеспечения, руководства пользователя, технические отчеты и оценки, обоснования, диаграммы и диаграммы и т. д.

Четкое и лаконичное письмо очень важно, потому что часто это основной способ общения между заинтересованными сторонами. Во всех случаях письменные продукты разработки программного обеспечения должны быть написаны так, чтобы они были доступны, понятны и актуальны для предполагаемой(ых) аудитории(ей).

3.3 Командное и групповое общение

[3 , с1с6.8] [4 , с22с3] [5 , с27с1] [9 , с10с4]

Эффективное общение между командой и членами группы имеет важное значение для совместной разработки программного обеспечения. Необходимо консультироваться с заинтересованными сторонами, принимать решения и разрабатывать планы. Чем больше количество членов команды и группы, тем больше потребность в общении.

Однако количество каналов связи увеличивается квадратично с добавлением каждого члена команды. Кроме того, члены команды вряд ли будут общаться с кем-либо, кого считают удаленным от них более чем на две степени (уровня). Эта проблема может быть более серьезной, когда усилия или организации по разработке программного обеспечения разбросаны по национальным и континентальным границам.

Некоторое общение может осуществляться в письменной форме. Документация по программному обеспечению является обычной заменой прямого взаимодействия. Электронная почта — это другое, но, хотя она и полезна, ее не всегда достаточно; также, если кто-то отправляет слишком много сообщений, становится трудно идентифицировать важную информацию. Все чаще организации используют корпоративные инструменты для совместной работы для обмена информацией. Кроме того, использование электронных хранилищ информации, доступных для всех членов команды, для организационных политик, стандартов, общих инженерных процедур и информации по конкретному проекту, может быть наиболее полезным.

Некоторые команды разработчиков программного обеспечения сосредотачиваются на личном взаимодействии и способствуют такому взаимодействию путем организации офисного пространства. Хотя частные офисы повышают индивидуальную производительность, совместное размещение членов команды в физических или виртуальных формах и предоставление общих рабочих мест важно для совместных усилий.

3.4 Навыки презентации

[3 , c1s5] [4 , c22] [9 , c10s7–c10s8]

Инженеры-программисты полагаются на свои навыки презентации в процессе жизненного цикла программного обеспечения. Например, на этапе разработки требований к программному обеспечению инженеры-программисты могут ознакомить клиентов и коллег с требованиями к программному обеспечению и провести формальные обзоры требований (см. Обзоры требований в КА «Требования к программному обеспечению»). Во время и после проектирования программного обеспечения, создания программного обеспечения и обслуживания программного обеспечения инженеры-программисты проводят обзоры, обзоры продуктов (см. Обзоры и аудиты в КА качества программного обеспечения) и обучение. Все это требует способности представлять техническую информацию группам и запрашивать идеи или отзывы.

Таким образом, способность инженера-программиста эффективно передавать концепции в презентации влияет на принятие продукта, управление им и поддержку клиентов; это также влияет на способность заинтересованных сторон понимать и помогать в работе над продуктом. Эти знания должны быть заархивированы в виде слайдов, описаний знаний, технических документов и любых других материалов, используемых для создания знаний.

ДАЛЬНЕЙШИЕ ЧТЕНИЯ

Джеральд М. Вайнберг, *Психология компьютерного программирования* [10].

Это была первая крупная книга, в которой программирование рассматривалось как индивидуальное и групповое усилие, и она стала классикой в этой области.

Кинни и Ланге, П.А., *Закон об интеллектуальной собственности для бизнес-юристов* [11].

В этой книге рассматриваются законы об интеллектуальной собственности в США. В нем не только говорится о том, что такое закон об ИС; это также объясняет, почему он выглядит именно так.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

[1] Ф. Ботт и др., *Профессиональные вопросы разработки программного обеспечения*, 3-е изд., Тейлор и Фрэнсис, 2000.

- [2] Энциклопедический словарь Merriam-Webster, 11-е изд., 2003 г.
- [3] Г. Воланд, *Проектирование по дизайну*, 2-е изд., Прентис Холл, 2003.
- [4] И. Соммервилль, *Разработка программного обеспечения*, 9-е изд., Addison-Wesley, 2011.
- [5] С. МакКоннелл, *Code Complete*, 2-е изд., Microsoft Press, 2004.
- [6] Объединенная рабочая группа IEEE CS/ACM по этике и профессиональной практике разработки программного обеспечения, «Кодекс этики и профессиональной практики разработки программного обеспечения (версия 5.2)», 1999 г., www.acm.org/serving/se/code.htm.
- [7] Дж. В. Мур, *Дорожная карта разработки программного обеспечения: руководство*, основанное на стандартах, издательство Wiley-IEEE Computer Society Press, 2006.
- [8] С. Токи, *Окупаемость программного обеспечения: максимизация окупаемости инвестиций в программное обеспечение*, Addison-Wesley, 2004.
- [9] RE Fairley, *Управление и руководство программными проектами*, Wiley-IEEE Computer Society Press, 2009.
- [10] GM Weinberg, *Психология компьютерного программирования: Silver Anniversary Edition*, Dorset House, 1998.
- [11] Кинни и Ланге, Пенсильвания, *Закон об интеллектуальной собственности для бизнес-юристов*, Thomson West, 2013.

Retrieved from "http://swbokwiki.org/index.php?title=Chapter_11:_Software_Engineering_Professional_Practice&oldid=875"

-
- Последнее изменение этой страницы состоялось 29 августа 2015 г., в 15:52.