

Содержание:

image not found or type unknown



Введение

Программная инженерия (англ. software engineering) — приложение систематического, дисциплинированного, измеримого подхода к разработке, функционированию и сопровождению программного обеспечения, а также исследованию этих подходов; то есть, приложение дисциплины инженерии к программному обеспечению.

Разработка программного обеспечения является достаточно молодой отраслью инженерной науки. С 90-х гг. XX в. Британское сообщество вычислительной техники (British Computer Society) стало присваивать разработчикам программ звание инженера, в США подобное звание появилось с 1998 г. И сегодня имеет смысл вести речь об инженерии программного обеспечения, или программной инженерии (научной дисциплине, охватывающей все аспекты жизненного цикла программного продукта), а также о специалистах в этой области — программных инженерах.

Цель современных технологий программной инженерии

Цель состоит в обеспечении эффективности всего жизненного цикла комплексов программ для ЭВМ в различных проблемно-ориентированных областях. В понятие современной технологии включается совокупность методов и инструментальных средств автоматизации, а также технологические процессы, обеспечивающие жизненный цикл сложных ПС с заданными функциональными и конструктивными характеристиками качества. Для этого рекомендуется использовать наиболее эффективные и совершенные методы проектирования и проводить комплексную автоматизацию ЖЦ ПС. Целеустремленная деятельность разработчиков-поставщиков должна быть направлена на удовлетворение требований заказчиков и пользователей программных продуктов при их применении по прямому

назначению.

Эта деятельность регламентируется рядом методов и стандартов, которые являются компонентами технологического обеспечения сложных ПС в течение их жизненного цикла. Их применение предполагает высокую дисциплину коллектива специалистов, использование им методик, стандартов, типовых нормативных документов и средств автоматизации разработки, которые регламентируют порядок организации и проведения работ по выполнению технологических операций, направленных на получение, в имеющихся организационно-технических условиях, готового программного продукта с заданными функциями и качеством.

Системные основы современных технологий программной инженерии

Методической основой технологии, регламентирующей деятельность специалистов, является типовой технологический процесс. Он отражается набором этапов и операций в последовательности их выполнения и взаимосвязи, обеспечивающих ведение работ на всех стадиях от инициирования проекта и подготовки технического задания до завершения испытаний или применения версии ПС. В современных технологиях объединяются методы непосредственной разработки программ и данных с методами обеспечения качества и организации управления их созданием с учетом технологических и человеческих факторов.

Индустриализация технологий программной инженерии базируется на стандартизации процессов разработки программ, их структурного построения и интерфейсов с операционной и внешней средой. Для этого с самого начала разработки должны определяться состав и этапы работ, необходимые для достижения конечной цели, а также требуемые для их выполнения ресурсы. Технические и управленческие проверки, анализ качества результатов промежуточных работ и компонентов, а также корректности их взаимосвязей должны обеспечивать руководителям и всем разработчикам уверенность достижения требуемого конечного результата проекта.

Значительные достижения в развитии и применении современных методов и технологии обеспечения крупномасштабных проектов ПС сосредоточены в методологии CMM (Capability Maturity Model – система и модель для оценки зрелости) комплекса технологических процессов жизненного цикла ПС, а также в её последующем развитии в CMMI. Она основана на формализации и использовании пяти уровней зрелости технологий поддержки ЖЦ ПС, которые также определяют потенциально возможное качество создаваемых на предприятии комплексов

программ.

Программная инженерия

С развитием языков программирования развивались и методики, применяемые для создания программного обеспечения, в результате чего в конце 90-х гг. XX в. появилось направление «программная инженерия». Компьютерное сообщество IEEE к 2004 г. сформировало Руководство к своду знаний по программной инженерии.

Согласно SWEBOOK программная инженерия включает в себя следующие десять областей знаний:

- 1) требования (software requirements);
- 2) проектирование (software design);
- 3) конструирование (software construction);
- 4) тестирование (software testing);
- 5) поддержка и эксплуатация (software maintenance);
- 6) конфигурационное управление (software configuration management);
- 7) управление инженерной деятельностью (software engineering management);
- 8) процессы инженерной деятельности (software engineering process);
- 9) инженерные инструменты и методы (software engineering tools and methods);
- 10) качество (software quality).

Вся деятельность разработчиков программного обеспечения осуществляется в рамках соответствующего проекта. Как и любой проект, программный проект реализуется в условиях ограниченности ресурсов, в качестве которых выступают стоимость проекта и его длительность. При этом создаваемый программный продукт должен быть наделен заявленной функциональностью и отвечать требованиям качества. Исходя из этого основная цель программных инженеров — это создание качественного программного продукта в пределах установленных заранее временных и финансовых рамок.

Для обеспечения требуемого уровня качества программного продукта используются следующие методы.

Инспектирование — команда разработчиков работает в направлении достижения качества на протяжении всего периода реализации проекта посредством коллективного рассмотрения разрабатываемого программного обеспечения и его

составляющих на предмет соответствия заданным требованиям или показателям качества.

Формальные методы — математические методики доказательства адекватности созданной программы установленным требованиям, применяемые выборочно.

Тестирование — проверка соответствия функциональных возможностей программы заявленным требованиям, осуществляемая на уровне модуля и целого приложения, а также проверка корректной работы программы в различных условиях.

Количественные характеристики качества отражаются в метриках, приведенных в соответствующих стандартах.

Специальные методы управления проектом — методы, используемые в процессе разработки для контроля стоимости и сроков реализации проекта, а также для грамотного управления версиями, документацией, конфигурациями и прочими артефактами.

Качественный программный продукт должен быть не только функциональным, удобным и адекватно использующим ресурсы вычислительной системы, но и отвечать установленным требованиям качества.

Надежность программной инженерии

Целью инженерии надежности программных систем является прогнозирование надежности программного обеспечения на основе статистических методов.

Проблема заключается в том, что программное обеспечение не отказывает, а выдает заранее определенные правильные или ошибочные выходные данные. Поэтому в основе SRE лежит предположение, что программное обеспечение действует не случайным образом, а конфигурация системы и вид операции могут рассматриваться как случайные условия.

SRE применяют в процессе испытаний при принятии решения о прекращении испытаний или для прогнозирования надежности при эксплуатации. Обычно данные отбирают в группах, например количество отказов за указанное время, поскольку очень трудно получить реальные наработки для отказов.

Большинство прикладных программ основаны на предположении, что программная ошибка может быть описана негомогенным процессом Пуассона. Это означает, что

программные ошибки происходят в статистически независимые моменты времени. Нарботки подчиняются экспоненциальному распределению, а интенсивность отказов изменяется во времени. Обычно используют убывающую интенсивность отказов. Это означает, что ошибки, как только они выявлены, эффективно устраняются без введения новых ошибок. Главная цель SRE заключается в том, чтобы определить форму функции интенсивности отказов и оценить ее параметры по наблюдаемым данным. Как только функция интенсивности отказов определена, могут быть найдены такие показатели надежности как:

- общее количество отказов;
- количество остающихся отказов;
- время до следующего отказа;
- остаточное время испытаний;
- максимальное количество отказов (относительно срока службы).

Другие подходы принимают во внимание архитектуру программного обеспечения, его функциональные модули, модель их взаимодействия. Затем данные отбирают и определяют оценки для модулей.

Ключевые элементы

- определение показателей надежности и целей;
- определение используемой модели надежности программного обеспечения;
- отбор данных об отказах;
- валидация модели;
- прогноз показателей надежности по данным.

Достоинства

- программное обеспечение может быть рассмотрено при прогнозировании надежности;

Ограничения

- сбор данных о надежности программного обеспечения может быть трудным, так как качество результатов определяется качеством собранных данных;
- нет подхода для выбора функций интенсивности отказов. Имеется искушение выбрать модель интенсивности отказов, которой данные соответствуют больше всего вместо априорного выбора модели;

- теоретическая основа негомогенного процесса Пуассона намного слабее, чем в случае прогнозирования надежности аппаратных средств.

Заключение

Таким образом, программная инженерия - приложение систематического, дисциплинированного, измеримого подхода к разработке, функционированию и сопровождению программного обеспечения. Методической основой технологии, регламентирующей деятельность специалистов, является типовой технологический процесс. Он отражается набором этапов и операций в последовательности их выполнения и взаимосвязи,

Индустриализация технологий программной инженерии базируется на стандартизации процессов разработки программ, их структурного построения и интерфейсов с операционной и внешней средой.

Большинство прикладных программ основаны на предположении, что программная ошибка может быть описана негомогенным процессом Пуассона. Это означает, что программные ошибки происходят в статистически независимые моменты времени. Ошибки, как только они выявлены, эффективно устраняются без введения новых ошибок.

СПИСОК ЛИТЕРАТУРЫ

1. [1.https://studref.com/320261/informatika/osnovnye_ponyatiya](https://studref.com/320261/informatika/osnovnye_ponyatiya)
2. [2.http://citforum.ru/SE/lipaev/](http://citforum.ru/SE/lipaev/)
3. [3.http://sewiki.ru/](http://sewiki.ru/)