

Лекция

Понятие жизненного цикла ПО

В качестве примеров деятельностей, которые нужно проводить для построения программной системы, можно привести **проектирование** — выделение отдельных модулей и определение связей между ними с целью минимизации зависимостей между частями проекта и достижения лучшей его обозримости в целом, **кодирование** — разработку кода отдельных модулей, разработку пользовательской документации, которая необходима для достаточно сложной системы.

Весь период существования ПО, связанный с подготовкой к его разработке, разработкой, использованием и модификациями, начиная с того момента, когда принимается решение разработать/приобрести/собрать из имеющихся компонентов новую систему или приходит сама идея о необходимости программы определенного рода, до того момента, когда полностью прекращается всякое ее использование, называют **жизненным циклом ПО**.

В ходе жизненного цикла ПО проходит через анализ предметной области, сбор требований, проектирование, кодирование, тестирование, сопровождение и др. **виды деятельности**. Каждый вид представляет собой достаточно однородный набор действий, выполняемых для решения одной задачи или группы тесно связанных задач в рамках разработки и поддержки эксплуатации ПО.

При этом создаются и перерабатываются различного рода **артефакты** — создаваемые человеком информационные сущности, документы в достаточно общем смысле, участвующие в качестве входных данных и получающиеся в качестве результатов различных деятельностей. Примерами артефактов являются: модель предметной области, описание требований, техническое задание, архитектура системы, проектная документация на систему в целом и на ее компоненты, прототипы системы и компонентов, собственно, исходный код, пользовательская документация, документация администратора системы, руководство по развертыванию, база пользовательских запросов, план проекта, и пр.

На различных этапах в создание и эксплуатацию ПО вовлекаются люди, выполняющие различные **роли**. Каждая роль может быть охарактеризована как абстрактная группа заинтересованных лиц, участвующих в деятельности по созданию и эксплуатации системы и решающих одни и те же задачи или имеющих одни и те же интересы по отношению к ней. Примерами ролей являются: бизнес-аналитик, инженер по требованиям, архитектор, проектировщик пользовательского интерфейса, программист-кодировщик, технический писатель, тестировщик, руководитель проекта по разработке, работник отдела продаж, конечный пользователь, администратор системы, инженер по поддержке и т.п.

Похоже, что общую структуру жизненного цикла любого ПО задать невозможно, поскольку она существенно зависит от целей, для которых это ПО разрабатывается или приобретается, и от решаемых им задач. Структура жизненного цикла будет существенно разной у программы для форматирования кода, которая сначала делалась программистом для себя, а впоследствии была признана перспективной в качестве продукта и переработана, и у комплексной системы автоматизации предприятия, которая с самого начала задумывалась как таковая.

Определяют основные элементы структуры жизненного цикла в виде **модели жизненного цикла ПО**. Модель жизненного цикла ПО выделяет конкретные наборы видов деятельности (обычно разбиваемых на еще более мелкие активности), артефактов, ролей и их взаимосвязи, а также дает рекомендации по организации процесса в целом. Эти рекомендации включают ответы на вопросы о том, какие артефакты являются входными данными у каких видов деятельности, а какие появляются в качестве результатов, какие роли вовлечены в различные деятельности, как различные деятельности связаны друг с другом, каковы критерии качества полученных результатов, как оценить степень соответствия различных артефактов общим задачам проекта и когда можно переходить от одной деятельности к другой.

Жизненный цикл ПО является составной частью жизненного цикла программно-аппаратной системы, в которую это ПО входит. Поэтому часто различные его аспекты рассматриваются в связи с элементами жизненного цикла системы в целом.

Существует набор стандартов, определяющих различные элементы в структуре жизненных циклов ПО и программно-аппаратных систем. В качестве основных таких элементов выделяют **технологические процессы** — структурированные наборы деятельностей, решающих некоторую общую задачу или связанную совокупность задач, такие, как процесс сопровождения ПО, процесс обеспечения качества, процесс разработки документации и пр. Процессы могут определять разные этапы жизненного цикла и увязывать их с различными видами деятельностей, артефактами и ролями заинтересованных лиц.

Стандарты жизненного цикла

Чтобы получить представление о возможной структуре жизненного цикла ПО, обратимся сначала к соответствующим стандартам, описывающим технологические процессы. Международными организациями, такими, как:

- IEEE — читается «ай-трипл-и», Institute of Electrical and Electronic Engineers, Институт инженеров по электротехнике и электронике;
- ISO — International Standards Organization, Международная организация по стандартизации;
- EIA — Electronic Industry Association, Ассоциация электронной промышленности;
- IEC — International Electrotechnical Commission, Международная комиссия по электротехнике;

а также некоторыми национальными и региональными институтами и организациями (в основном, американскими и европейскими, поскольку именно они оказывают наибольшее влияние на развитие технологий разработки ПО во всем мире):

- ANSI — American National Standards Institute, Американский национальный институт стандартов;
- SEI — Software Engineering Institute, Институт программной инженерии;
- ECMA — European Computer Manufacturers Association, Европейская ассоциация производителей компьютерного оборудования;

разработан набор стандартов, регламентирующих различные аспекты жизненного цикла и вовлеченных в него процессов. Список и общее содержание этих стандартов представлены ниже.

Группа стандартов ISO

• **ISO/IEC 12207 Standard for Information Technology — Software Life Cycle Processes** [1] (процессы жизненного цикла ПО, есть его российский аналог **ГОСТ Р-1999** [2]). Определяет общую структуру жизненного цикла ПО в виде 3-х ступенчатой модели, состоящей из процессов, видов деятельности и задач. Стандарт описывает вводимые элементы в терминах их целей и результатов, тем самым задавая неявно возможные взаимосвязи между ними, но не определяя четко структуру этих связей, возможную организацию элементов в рамках проекта и метрики, по которым можно было бы отслеживать ход работ и их результативность.

Самыми крупными элементами являются *процессы жизненного цикла ПО (lifecycle processes)*. Всего выделено 18 процессов, которые объединены в 4 группы.

Основные процессы	Поддерживающие процессы	Организационные процессы	Адаптация
Приобретение ПО; Передача ПО (в использование); Разработка ПО; Эксплуатация ПО; Поддержка ПО	Документирование; Управление конфигурациями; Обеспечение качества; Верификация; Валидация; Совместные экспертизы; Аудит; Разрешение проблем	Управление проектом; Управление инфраструктурой; Усовершенствование процессов; Управление персоналом	Адаптация описываемых стандартом процессов под нужды конкретного проекта

Таблица 1. Процессы жизненного цикла ПО по ISO 12207.

Процессы строятся из отдельных *видов деятельности (activities)*. Стандартом определены 74 вида деятельности, связанной с разработкой и поддержкой ПО. Здесь мы упомянем только некоторые из них.

о Приобретение ПО включает такие деятельности, как инициация приобретения, подготовка запроса предложений, подготовка контракта, анализ поставщиков, получение ПО и завершение приобретения.

о Разработка ПО включает развертывание процесса разработки, анализ системных требований, проектирование (программно-аппаратной) системы в целом, анализ требований к ПО, проектирование архитектуры ПО, детальное проектирование, кодирование и отладочное тестирование, интеграцию ПО, квалификационное тестирование ПО, системную интеграцию, квалификационное тестирование системы, развертывание (установку или инсталляцию) ПО, поддержку процесса получения ПО.

о Поддержка ПО включает развертывание процесса поддержки, анализ возникающих проблем и необходимых изменений, внесение изменений, экспертизу и передачу измененного ПО, перенос ПО с одной платформы на другую, изъятие ПО из эксплуатации.

о Управление проектом включает запуск проекта и определение его рамок, планирование, выполнение проекта и надзор за его выполнением, экспертизу и оценку проекта, свертывание проекта.

Каждый вид деятельности нацелен на решение одной или нескольких *задач (tasks)*. Всего определено 224 различные задачи. Например:

о Развертывание процесса разработки состоит из определения модели жизненного цикла, документирования и контроля результатов отдельных работ, выбора используемых стандартов, языков и инструментов и пр.

о Перенос ПО между платформами состоит из разработки плана переноса, оповещения пользователей, выполнения анализа произведенных действий и пр.

• **ISO/IEC 15288 Standard for Systems Engineering — System Life Cycle Processes** [3] (процессы жизненного цикла систем). Отличается от предыдущего нацеленностью на рассмотрение программно-аппаратных систем в целом. В данный момент продолжается работа по приведению этого стандарта в соответствие с предыдущим. ISO/IEC 15288 предлагает похожую схему рассмотрения жизненного цикла системы в виде набора процессов. Каждый процесс описывается набором его *результатов (outcomes)*, которые достигаются при помощи различных видов деятельности.

Всего выделено 26 процессов, объединяемых в 5 групп.

Процессы выработки соглашений	Процессы уровня организации	Процессы уровня проекта	Технические процессы	Специальные процессы
Приобретение системы; Поставка системы	Управление окружением; Управление инвестициями; Управление процессами; Управление ресурсами; Управление качеством	Планирование; Оценивание; Мониторинг; Управление рисками; Управление конфигурацией; Управление информацией; Выработка решений	Определение требований; Анализ требований; Проектирование архитектуры; Реализация; Интеграция; Верификация; Валидация; Передача в использование; Эксплуатация; Поддержка; Изъятие из эксплуатации	Адаптация описываемых стандартом процессов под нужды конкретного проекта

Таблица 2. Процессы жизненного цикла систем по ISO 15288.

Помимо процессов, определено 123 различных результата и 208 видов деятельности, нацеленных на их достижение. Например, определение требований имеет следующие результаты.

Отношения заказчик-поставщик	Процессы уровня организации	Процессы уровня проекта	Инженерные процессы	Процессы поддержки
Приобретение ПО; Составление контракта; Определение нужд заказчика; Проведение совместных экспертиз и аудитов; Подготовка к передаче; Поставка и развертывание; Поддержка эксплуатации; Предоставление услуг; Оценка удовлетворенности заказчиков	Развитие бизнеса; Определение процессов; Усовершенствование процессов; Обучение; Обеспечение переносимости; Обеспечение инструментами; Обеспечение среды для работы	Планирование жизненного цикла; Планирование проекта; Построение команды; Управление требованиями; Управление качеством; Управление рисками; Управление ресурсами и графиком работ; Управление подрядчиками	Выделение системных требований и проектирование системы в целом; Выделение требований к ПО; Проектирование ПО; Реализация, интеграция и тестирование ПО; Интеграция и тестирование системы; Сопровождение системы и ПО	Разработка документации; Управление конфигурацией; Обеспечение качества; Разрешение проблем; Проведение экспертиз

Таблица 3. Процессы жизненного цикла ПО и систем по ISO 15504.

- о Должны быть поставлены технические задачи, которые предстоит решить.
- о Должны быть сформулированы системные требования.

Деятельности в рамках этого процесса следующие.

- о Определение границ функциональности системы.

- o Определение функций, которые необходимо поддерживать.
- o Определение критериев оценки качества при использовании системы.
- o Анализ и выделение требований по безопасности.
- o Анализ требований защищенности.
- o Выделение критических для данной системы аспектов качества и требований к ним.
- o Анализ целостности системных требований.
- o Демонстрация прослеживаемости требований.
- o Фиксация и поддержка набора системных требований.

• **ISO/IEC 15504 (SPICE) Standard for Information Technology — Software Process Assessment [4]** (оценка процессов разработки и поддержки ПО). Определяет правила оценки процессов жизненного цикла ПО и их возможностей, опирается на модель СММІ (см. ниже) и больше ориентирован на оценку процессов и возможностей их улучшения. В качестве основы для оценки процессов определяет некоторую базовую модель, аналогичную двум описанным выше. В ней выделены категории процессов, процессы и виды деятельности.

Например, приобретение ПО включает такие виды деятельности, как определение потребности в ПО, определение требований, подготовку стратегии покупки, подготовку запроса предложений, выбор поставщика.

Группа стандартов IEEE

• **IEEE 1074-1997 — IEEE Standard for Developing Software Life Cycle Processes [5]** (стандарт на создание процессов жизненного цикла ПО). Нацелен на описание того, как создать специализированный процесс разработки в рамках конкретного проекта. Описывает ограничения, которым должен удовлетворять любой такой процесс, и, в частности, общую структуру процесса разработки. В рамках этой структуры определяет основные виды деятельности, выполняемых в этих процессах и документы, требующиеся на входе и возникающие на выходе этих деятельностей. Всего рассматриваются 5 подпроцессов, 17 групп деятельностей и 65 видов деятельности. Например, подпроцесс разработки состоит из групп деятельностей по выделению требований, по проектированию и по реализации. Группа деятельностей по проектированию включает архитектурное проектирование, проектирование баз данных, проектирование интерфейсов, детальное проектирование компонентов.

• **IEEE/EIA 12207-1997 — IEEE/EIA Standard: Industry Implementation of International Standard ISO/IEC 12207:1995 Software Life Cycle Processes [6-8]** (промышленное использование стандарта ISO/IEC 12207 на процессы жизненного цикла ПО). Аналог ISO/IEC 12207, сменил ранее использовавшиеся стандарты J-Std-016-1995 EIA/IEEE Interim Standard for Information Technology — Software Life Cycle Processes — Software Development Acquirer-Supplier Agreement (промежуточный стандарт на

процессы жизненного цикла ПО и соглашения между поставщиком и заказчиком ПО) и стандарт министерства обороны США MIL-STD-498.

Группа стандартов СММ, разработанных SEI

• **Модель зрелости возможностей СММ (Capability Maturity Model)** [9,10] предлагает унифицированный подход к оценке возможностей организации выполнять задачи различного уровня. Для этого определяются 3 уровня элементов: *уровни зрелости организации (maturity levels)*, *ключевые области процесса (key process areas)* и *ключевые практики (key practices)*. Чаще всего под моделью СММ имеют в виду модель уровней зрелости. В настоящий момент СММ считается устаревающей и сменяется моделью СММІ (см. ниже).

о Уровни зрелости. СММ описывает различные степени зрелости процессов в организациях, определяя 5 уровней организаций.

□ Уровень 1, начальный (initial). Организации, разрабатывающие ПО, но не имеющие осознанного процесса разработки, не производящие планирования и оценок своих возможностей.

□ Уровень 2, повторяемый (repeatable). В таких организациях ведется учет затрат ресурсов и отслеживается ход проектов, установлены правила управления проектами, основанные на имеющемся опыте.

□ Уровень 3, определенный (defined). В таких организациях имеется принятый, полностью документированный, соответствующий реальному положению дел и доступный персоналу процесс разработки и сопровождения ПО. Он должен включать как управленческие, так и технические подпроцессы, а также обучение сотрудников работе с ним.

□ Уровень 4, управляемый (manageable). В этих организациях, помимо установленного и описанного процесса, используются измеримые показатели качества продуктов и результативности процессов, которые позволяют достаточно точно предсказывать объем ресурсов (времени, денег, персонала), необходимый для разработки продукта с определенным качеством.

□ Уровень 5, совершенствующийся (optimizing). В таких организациях, помимо процессов и методов их оценки, имеются методы определения слабых мест, определены процедуры поиска и оценки новых методов и техник разработки, обучения персонала работе с ними и их включения в общий процесс организации в случае повышения эффективности производства.

о Ключевые области процесса. Согласно СММ, уровни зрелости организации можно определять по использованию четко определенных техник и процедур, относящихся к различным ключевым областям процесса. Каждая такая область представляет собой набор связанных видов деятельности, которые нацелены на достижение целей, существенных для общей оценки результативности технологического процесса. Всего выделяется 18 областей. Множество областей, которые должны

поддерживаться организацией, расширяется при переходе к более высоким уровням зрелости.

- К первому уровню не предъявляется никаких требований.
- Организации второго уровня зрелости должны поддерживать управление требованиями, планирование проектов, надзор за ходом проекта, управление подрядчиками, обеспечение качества ПО, управление конфигурацией.

- Организации третьего уровня должны, помимо деятельности второго уровня, поддерживать проведение экспертиз, координацию деятельности отдельных групп, разработку программного продукта, интегрированное управление разработкой и сопровождением, обучение персонала, выработку и поддержку технологического процесса организации, контроль соблюдения технологического процесса организации.

- К деятельности организаций четвертого уровня добавляются: управление качеством ПО и управление процессом, основанное на измеримых показателях.

- Организации пятого уровня зрелости должны дополнительно поддерживать управление изменениями процесса, управление изменениями используемых технологий и предотвращение дефектов.

о Ключевые практики. Ключевые области процесса описываются с помощью наборов ключевых практик. Ключевые практики классифицированы на несколько видов: обязательства (commitments to perform), возможности (abilities to perform), деятельности (activities performed), измерения (measurements and analysis) и проверки (verifying implementations)