## Разработка требований

***Разработка требований*** включает в себя мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований.

Различают четыре основных этапа процесса разработки требований:

1. анализ технической осуществимости создания системы;
2. формирование и анализ требований;
3. специфицирование требований и создание соответствующей документации;
4. аттестация требований и процесс управления изменениями системных требований.

Э т ап 1 . А н ал и з т ех н и ч еск о й о с у щ е ст в и м о с т и со з д ан и я си ст ем ы .

Процесс разработки требований начинается с описания и анализа создаваемой системы и ее назначения. Для этого определяются источники информации (менеджеры отделов, разработчики программного обеспечения, технологи, конечные пользователи и т.д.), затем производится сбор и анализ информации о будущей системе, определяются решаемые задачи и выбираются технологии реализации.

В соответствии с ГОСТ Р ИСО/МЭК 12207:

1) заказчик инициирует процесс заказа, описывая концепцию или потребность в заказе, разработке или модернизации системы, программного продукта или программной услуги;

2) заказчик определяет и анализирует требования к системе.

Требования к системе должны охватывать функциональные, коммерческие, организационные и потребительские аспекты системы, а также требования безопасности, защиты и другие критические требования наряду с требованиями к проектированию, тестированию и соответствующим стандартам и процедурам.

В результате создается итоговый документ, содержащий требования, отвечающие бизнес-целям организации-заказчика и организации- разработчика, результаты анализа осуществимости проекта в пределах выделенных средств и сроков, а также результаты анализа возможности интеграции с существующими эксплуатируемыми системами заказчика. Должен быть предложен обоснованный бюджет и график работ по созданию системы. Важно подчеркнуть, что анализ осуще- ствимости выработанных требований необходим в целях минимизации риска срыва всего проекта.

Э т а п 2 . Ф орм и ро в а н и е и а н а л и з т ре б ов а н и й . На этом этапе

команда разработчиков ПО совместно с заказчиком и конечными пользователями системы определяют области применения, системные сервисы, режимы работы системы и ее характеристики, аппаратные ограничения и т.д.

В формировании и анализе требований принимают участие различные группы людей: пользователи, инженеры-разработчики, бизнес-менеджеры, специалисты в предметной области. В связи с этим сложно сформулировать общие требования, потому что каждая сторона выражает свою точку зрения и интересы. К тому же многие участники процесса могут выдвигать трудноосуществимые требования, так как не могут адекватно оценить стоимость их реализации.

Одним из результатов анализа требований является достижение понимания и согласия относительно функциональных возможностей ПО. Как правило, процесс формирования и анализа требований проходит через ряд этапов.

1. ***Анализ предметной области.*** Аналитики должны изучить предметную область и среду, в которой будет эксплуатироваться система.
2. ***Сбор требований.*** Это процесс формирования требований, в котором участвуют все заинтересованные лица. При этом продолжается анализ предметной области.
3. ***Классификация требований.*** На этом этапе весь набор требований преобразуется в логически связанные группы.
4. ***Разрешение противоречий.*** Требования лиц, занятых в процессе их формирования, могут быть противоречивыми. На этом этапе определяются и разрешаются обнаруженные противоречия.
5. ***Назначение приоритетов.*** На этом этапе требованиям назначаются приоритеты в соответствии с важностью задач для бизнес- целей заказчика.
6. ***Проверка требований.*** На этом этапе определяется полнота, последовательность и непротиворечивость требований.

К формированию и анализу требований не существует универсального подхода.

Ниже приводятся три метода, облегчающих процесс формирования требований.

1. ***Метод, основанный на множестве опорных точек зрения.*** При разработке требований выявляются общие интересы и точки зрения участников процесса создания ПО. При таком подходе происходит идентификация опорных точек зрения и соответствующих им сервисов. Один и тот же сервис может быть соотнесен с несколькими точками зрения. Присутствие сервисов, которым не было сопоставлено ни одной опорной точки зрения, означает, что на начальном этапе некоторые опорные точки зрения не были идентифицированы. Такой подход позволяет эффективно выявлять противоречия в требованиях, предложенных различными лицами.

Далее происходит обобщение мнений, связанных с каждой опорной точкой зрения, и разрешение возникших противоречий.

1. ***Метод построения сценариев.*** Как правило, воспринимать абстрактное описание системы гораздо сложнее, чем сценарий взаимодействия с ней. В процессе обсуждения сценариев формулируются новые требования и детализируются уже существующие. Обычно сценарий состоит из начального описания состояния системы, описания нормального протекания событий, описания исключительных ситуаций и способов их обработки, информации относительно других действий, которые можно осуществлять во время выполнения сценария, а также описания состояния системы после завершения сценария. Существуют различные мнения относительно наилучшего способа описания сценариев: некоторые предпочитают использовать диаграммы прецедентов (use case diagram), другие предпочитают не- формальное описание сценариев на естественном языке.
2. ***Этнографический метод.*** Учет социальных и организационных требований при создании ПО часто имеет большое значение для успеха системы на рынке. При использовании этнографического подхода разработчики требований погружаются в среду, в которой будет эксплуатироваться создаваемое ПО, с целью выявления неявных требований к системе, которые отражают реальные аспекты ее эксплуатации, а не формальные умозрительные процессы.

Этнографический подход позволяет детализировать требования для критических систем, чего не всегда можно добиться другими методами разработки требований. Однако этот метод ориентирован на конечного пользователя, он не может охватить всех требований предметной области и требований организационного характера, поэтому должен использоваться совместно с другими подходами.

В соответствии с ГОСТ Р ИСО/МЭК 12207 разработчик должен установить и

***документально оформить*** следующие требования к программным средствам:

* + функциональные и технические требования, включая производительность, физические характеристики и окружающие условия, под которые должно быть создано ПО;
  + требования к внешним интерфейсам ПО;
  + квалификационные требования;
  + требования безопасности;
  + требования защиты;
  + эргономические требования;
  + требования к определению данных и базе данных;
  + требования по вводу в действие и приемке поставляемого ПО на объектах эксплуатации и сопровождения;
  + требования к документации пользователя;
  + требования к эксплуатации ПО пользователем;
  + требования к обслуживанию пользователя.

Э т а п 3 . С пе циф ицир о в а ние т р еб о в ан и й и со з д а ние со о т в е т с т в у ю щ е й д о к у м ен т ац и и . Результатом анализа требований является официальный документ для разработчиков ПО, который в соответствии с ГОСТ

* 1. — 77 называют ***техническим заданием***  или ***спецификацией требований к программному обеспечению (Software Requirements Specification,SRS)***

Он содержит совокупность формализованных требований к ПО и используется как основа для разработки программной системы.

Ниже приводятся состав и содержание требований к АС в соответствии со стандартом ГОСТ 34.602 — 89. *1. Требования к системе в целом:*

* + - требования к структуре и функционированию системы;
    - требования к численности и квалификации персонала системы и режиму его работы;
    - показатели назначения;
    - требования к надежности;
    - требования безопасности;
    - требования к эргономике и технической эстетике;
    - требования к транспортабельности для подвижных АС;
    - требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
    - требования к защите информации от несанкционированного доступа;
    - требования по сохранности информации при авариях;
    - требования к защите от влияния внешних воздействий;
    - требования к патентной чистоте;
    - требования по стандартизации и унификации;
    - дополнительные требования.

1. Требования к функциям {задачам), выполняемым системой:
   * + по каждой подсистеме перечень функций и задач (в том числе обеспечивающих взаимодействие частей системы), подлежащих автоматизации;
     + при создании системы в две или более очереди — перечень функциональных подсистем, отдельных функций или задач, вводимых в действие в первой и последующих очередях;
     + временной регламент реализации каждой функции, задачи (или комплекса задач);
     + требования к качеству реализации каждой функции (задачи или комплекса задач), форме представления выходной информации, необходимой точности и времени выполнения, а также одновременности выполнения группы функций и достоверности выдачи результатов;
     + перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.
2. К *требованиям к видам обеспечения* относятся: требования к

математическому, информационному, лингвистическому, программному, техническому, метрологическому, организационному, методическому и другим видам обеспечения системы.

Для документирования требований, анализа системы и ее модификации могут использоваться ***модели системы.*** Модель является абстракцией системы, опускает детали и легче поддается анализу, чем любое другое представление этой системы. Она является связующим звеном между процессом анализа исходной задачи и процессом проектирования системы.

Модели, как правило, используются в качестве иллюстративного материала для представления системы в различных аспектах:

Внешнее представление, когда моделируется окружение или рабочая среда системы (*модели системного окружения).*

1. Описание поведения системы, когда моделируется ее поведение

*(поведенческие модели).*

1. Описание структуры системы, когда моделируется системная архитектура или структуры данных, обрабатываемых системой (*модели данных, объектные модели, модели наследования).*

Объектное моделирование соединяет в себе поведенческое и структурное моделирование. Такие структурные методы, как структурный анализ систем и объектно-ориентированный анализ, обеспечивают основу для детального моделирования системы как части процесса формулировки и анализа требований. Для поддержки структурных методов существуют различные ***CASE-средства,*** представ- ляющие собой пакет программных средств, который поддерживает отдельные этапы процесса разработки ПО. В этот пакет могут входить:

* + - редакторы диаграмм потоков данных;
    - средства проектирования, анализа, проверки ПО;
    - средства генерирования отчетов;
    - средства создания форм.

***Для разработки диаграмм, поясняющих требования, может использоваться*** унифицированный язык моделирования ( Unified Modeling Language, UML). ***Кроме того, могут применяться специализированные диаграммы, такие как*** диаграммы модели сущность-связь {entity-relationship diagram, ERD) ***и*** диаграммы потоков данных (Data Flow Diagrams, DFD).

Для формализованного описания системных требований может использоваться ***язык описания программ (Program Description Language, PDL).*** В качестве PDL может использоваться любой универсальный язык программирования.

Э т а п 4 . А т т е с т а ция т р е б о в а ний и пр о це с с у п р а в л е ния

и з м ен ен и я м и си ст ем н ы х т р еб о в ан и й .

Аттестация должна подтвердить, что разработанные на предыдущих этапах требования действительно определяют систему, необходимую заказчику. Проверка требований важна, так как ошибки в спецификации требований могут привести к последующей переделке системы, а следовательно, к большим материальным и временным затратам.

В соответствии с ГОСТ Р ИСО/МЭК 12207 — 99 разработчик должен оценить требования по следующим критериям (при этом результаты оценок должны быть документально оформлены):

* + - учет требований к системе и проекту системы;
    - внешняя согласованность с требованиями к системе;
    - внутренняя согласованность требований к ПО между собой;
    - тестируемость требований;
    - выполнимость программного проекта;
    - возможность эксплуатации и сопровождения.

После успешного проведения оценки должно быть документально зафиксировано состояние требований к ПО.

Существует ряд методов аттестации требований, которые можно использовать совместно или каждый в отдельности.

1. ***Обзор требований —*** это процесс просмотра системной спецификации для нахождения неточных описаний и ошибок.

Обзор требований может быть неформальным и формальным. Неформальный обзор — это простое обсуждение требований с большим количеством лиц, участвующих в их формировании. При формальном обзоре группа разработчиков должна обосновать причину включения каждого требования в спецификацию. При этом проверяется непротиворечивость требований и их полнота.

1. ***Прототипирование.*** Прототип системы может создаваться для уточнения неясных требований. Прототип предоставляется конечным пользователям и заказчику, которые могут провести с ним эксперименты, позволяющие уточнить требования.
2. ***Генерация тестовых сценариев.*** Тесты для требований разрабатываются как часть процесса аттестации, что позволяет обнаружить проблемы в спецификации. Если такие тесты сложно или невозможно разработать, то обычно это означает, что требования трудно выполнить, и поэтому необходимо их пересмотреть.