

1

Основы системной инженерии

Эта глава посвящена основам системной инженерии: ее истории и причинам ее использования. Кроме того, мы разберем основные понятия и терминологию, которая будет использоваться в книге. Это поможет вам лучше понимать содержимое других глав.

Краткая история системной инженерии

Можно сказать, что системная инженерия применяется с того момента, как человечество начало строить и разрабатывать сложные системы. Примеры таких систем — пирамиды Древнего Египта, а также простые каменные конструкции вроде хенджей¹, которые могли быть частью более крупной астрологической системы. Более того, человечество наблюдало за сложными системами, такими как Солнечная система, с тех пор, как древние греки впервые заметили движение планет и построили модель геоцентрической Вселенной.

Термин «системная инженерия» появился в начале XX века в Bell Laboratories в США. Применение системной инженерии можно проследить до времен Второй мировой войны, а впервые эту дисциплину начали преподавать в Массачусетском технологическом институте в 1950 году.

В 1960-х годах оформилась новая область наук с названием «теория систем». Основоположителем ее можно считать Людвиг фон Берталанфи, который занимался исследованием систем и дал ей первоначальное название «общая теория систем» (Bertalanffy, 1968).

Основная идея теории систем гласит: это *концептуальная основа, выстроенная на том принципе, что компоненты системы можно по-настоящему понять*

¹ Хендж — доисторическое круговое земляное сооружение с внутренним рвом и валом, часто использовавшееся вместе с каменными или деревянными конструкциями (например, Стоунхендж). — *Примеч. пер.*

в контексте их взаимодействия друг с другом и с иными системами, а не по отдельности (Wilkinson, 2011). Это важный момент для всей системной инженерии, поскольку элементы в системе или же сами системы никогда не рассматриваются сами по себе, а только во взаимосвязи с другими элементами или системами.

По мере усложнения систем становилось очевидно, что их разработка нуждается в новом подходе. На протяжении второй половины XX века эта потребность только росла. В ответ на это в 1990 году в США был основан *Национальный совет по системной инженерии* (National Council on Systems Engineering, NCOSE). В 1995 году он был переименован в *Международный совет по системной инженерии* (International Council on Systems Engineering, INCOSE). На данный момент это одна из самых авторитетных организаций в своей области, имеющая 70 представительств по всему миру.

Сегодня, когда сложность нашего мира и разрабатываемых систем растет все быстрее, необходимы строгие и надежные подходы, которые позволят справляться с этим высоким уровнем сложности. Системная инженерия — именно такой подход.

Определения

Начиная разговор о системной инженерии, важно четко понимать смысл используемых терминов. В этой дисциплине (как и во многих других) один термин редко имеет только одно точное определение. Это может приводить к проблемам в коммуникации — ключевом процессе успешной системной инженерии.

Во избежание этих проблем в текущей главе поясняются основные понятия и термины, которые будут использоваться в книге. Это позволит сформировать предметно-ориентированный язык. Там, где это возможно и целесообразно, терминология будет основываться на международных передовых практиках, в том числе стандартах, таких как ISO 15288 (ISO, 2015), чтобы обеспечить достоверность представленной здесь информации.

Что такое система

Начнем с базового понятия системной инженерии — *система*. Определений системы существует множество, и каждое зависит от контекста. Поэтому прежде всего определим несколько видов систем, типичных для системной инженерии.

Существует множество классификаций (таксономий) систем, и одна из наиболее популярных определена Питером Чеклендом (Checkland, 1999). Она показана на рис. 1.1.



Рис. 1.1. Пять типов систем по Чекленду

Вот более подробное описание этих систем.

- **Естественные системы** — открытые, их свойства неподвластны контролю человека. Примеры: погодные системы, природные экосистемы, окружающая среда, время и т. д.
- **Спроектированные физические системы** — обычно состоят из физических предметов, представляющих собой реальное материальное воплощение системы. Примеры: смартфоны, планшеты, вертолеты, автомобили, поезда, самолеты, космические корабли, телевизоры, камеры, мосты, компьютерные игры, спутники, домашняя техника. Список практически бесконечен.
- **Спроектированные абстрактные системы** — не представлены физическим предметом, но используются для понимания или объяснения идеи или концепции. Примеры: модели, уравнения, мысленные эксперименты и пр.
- **Системы человеческой деятельности** — существуют в человеческом обществе и широко распространены в реальном мире. Как правило, состоят из разных групп людей, взаимодействующих для достижения общей цели или задачи. Примеры: политические системы, социальные группы, услуги, основанные на взаимодействии людей, и т. д.
- **Трансцендентные системы** — выходят за пределы нашего текущего понимания. Примеры: божества, а также еще не изученные явления или задачи.

Это оптимальная классификация, дающая полное представление о разных типах систем. Она будет использоваться далее в книге. Важно понимать, что системная инженерия может применяться ко всем этим пяти типам.

Существуют системы, относящиеся сразу к нескольким типам. Например, транспортная система включает в себя транспортные средства (спроектированные физические системы), схемы и модели работы (спроектированные абстрактные системы), природную среду (естественная система) и управление транспортом (система человеческой деятельности). В реальной жизни сложные системы часто сочетают в себе несколько типов.

Характеристики системы

У всех систем есть общие характеристики, которые позволяют понимать и разрабатывать системы. Рассмотрим их подробнее.

Элементы

Любая система имеет свойственную ей структуру, и ее можно рассматривать как набор взаимодействующих *элементов* (рис. 1.2).

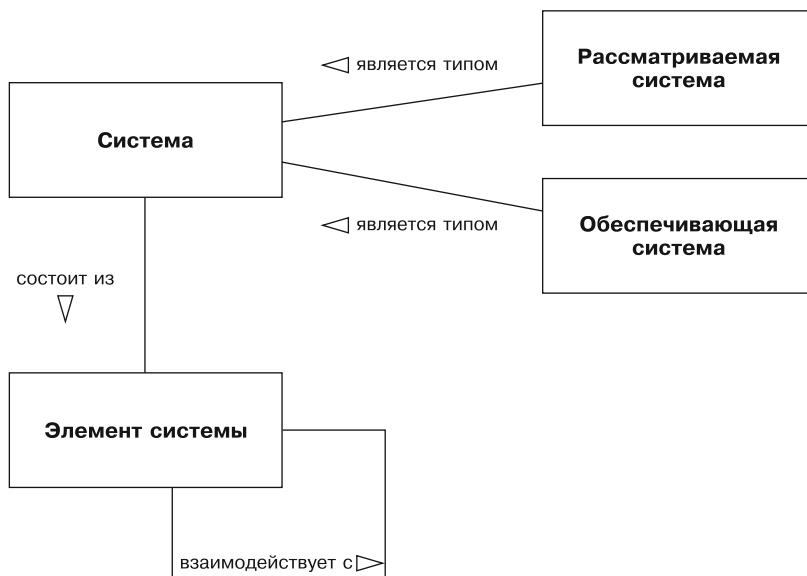


Рис. 1.2. Простая структура системы

Эта схема показывает, что система состоит из набора элементов и что есть два типа систем — *рассматриваемые* (system of interest) и *обеспечивающие* (enabling system). Рассматриваемая — это главная система, над которой ведется работа, а обеспечивающая — это любая система, которая взаимодействует с главной и помогает ей существовать.

Следует отметить, что структура системы на самом деле сложнее показанной выше, поскольку каждый элемент можно разбить на более мелкие составляющие, что, в свою очередь, приведет к созданию иерархии, состоящей из нескольких уровней. Поскольку мы лишь начали обсуждать эту тему, пока ограничимся системами с небольшим количеством уровней, чтобы объяснения оставались простыми. Позже, при более подробном рассмотрении систем, вы увидите иерархии со множеством уровней.

Следующий важный момент состоит в том, что элементы системы взаимодействуют друг с другом. Именно взаимодействие — одно из ключевых понятий, которое помогает понять, что такое настоящая система, и правильно применять принципы системной инженерии. Рассматривая любую систему или элемент, важно понимать, что они будут взаимодействовать с другими элементами, а не существовать изолированно. В системной инженерии все связано между собой, поэтому понимать, как взаимосвязаны элементы, так же важно, как и понимать, что они собой представляют.

Кроме того, взаимодействия элементов системы позволяют выявлять и определять интерфейсы между ними. Иметь четкое представление об этих интерфейсах критически важно, так как оно дает возможность точно определять все типы систем. При этом важно учитывать, какая информация или какие материалы (все, что не является информацией) передаются через эти интерфейсы.

Подробнее о структурах систем и интерфейсов мы поговорим в главе 3.

Заинтересованные стороны (стейкхолдеры)

Один из ключевых факторов, который важно учитывать при работе в области системной инженерии, — это наличие *заинтересованных сторон (стейкхолдеров)*. На рис. 1.3 показано, что стейкхолдер проявляет интерес к системе.



Рис. 1.3. Стейкхолдер и система

Под стейкхолдером понимается любое лицо, организация или объект, заинтересованные в системе.

Тема стейкхолдеров имеет несколько нюансов.

- При рассмотрении стейкхолдеров важна их роль, а не имя или название организации или предмета, связанного с ней. Например, возьмем человека

по имени Джон, у которого есть автомобиль. Джон как человек не стейкхолдер в отношении автомобиля. Скорее, им является роль, которую Джон играет при взаимодействии с автомобилем. Таким образом, в этом примере он играет несколько ролей заинтересованной стороны: *владелец, водитель, пассажир, плательщик, автомеханик* и т. д. Каждая из этих ролей будет по-разному воспринимать систему автомобиля. Поэтому важно рассматривать не самого Джона как человека, а роли стейкхолдера, которые он играет.

- Стейкхолдеры не обязательно люди. Заинтересованная сторона может быть организацией или чем-то еще. Например, при рассмотрении системы автомобиля роль стейкхолдера как *владельца* может исполнять Джон. Но автомобиль может принадлежать организации, в таком случае роль стейкхолдера берет на себя она, а не человек.
- Стейкхолдеры и люди, организации или объекты не соотносятся напрямую по принципу «один к одному». Например, один человек, Джон, может одновременно играть несколько ролей стейкхолдера. В свою очередь, одна и та же роль стейкхолдера может принадлежать сразу нескольким людям. Так, пассажиры, едущие в автомобиле вместе с водителем, выполняют одну и ту же роль стейкхолдера — пассажира.
- Стейкхолдеры находятся за пределами границ системы, как и обеспечивающие системы. Поскольку стейкхолдер определяется как любой объект, заинтересованный в системе, то можно сказать, что обеспечивающая система — это особый тип стейкхолдера, так как базовое определение у них совпадает.

Выявление стейкхолдеров — важная часть системной инженерии, так как в зависимости от своей роли на одну и ту же систему они будут смотреть по-разному. Это подводит нас к понятию контекста, которое мы рассмотрим чуть позже.

Атрибуты

Общие свойства любой системы можно описать, определив набор *атрибутов* (рис. 1.4).



Рис. 1.4. Атрибуты описывают свойства системы

Если помнить, что система состоит из элементов, то понятно, что эти атрибуты могут относиться и к элементам системы.

Атрибуты обычно представлены существительными, которые могут иметь несколько разных значений и быть конкретного предопределенного типа, а также измеряться в конкретных единицах. Вот примеры простых типов атрибутов.

- **Размеры**, такие как длина, ширина и высота, выраженные действительными числами, возможно, с миллиметрами как единицами измерения.
- **Вес**, выраженный действительным числом и измеряемый в килограммах как единицах.
- **Номер элемента** может быть целым числом и не иметь связанной с ним единицы измерения.
- **Имя** может быть буквой или текстом и не иметь связанной с ним единицы измерения.

Атрибуты могут иметь и более сложный тип.

- **Метка даты и времени** может быть набором простых типов, собранных вместе для создания более сложного типа. Например, сочетанием дня (целое число от 1 до 31), месяца (целое число от 1 до 12), года (целое число от 0000 и далее), часа (целое число от 1 до 24), минуты (целое число от 0 до 59) и секунды (целое число от 0 до 59).
- **Структуры данных** могут представлять собой целый аудио- или видеофайл, соответствующий определенному протоколу, например MP3, MP4 и т. д.

Полный набор возможных атрибутов почти бесконечен, поэтому приведенные выше списки служат лишь примером и не претендуют на полноту.

Границы

Каждая система будет иметь по крайней мере одну связанную с ней *границу*, которая помогает определить область действия этой системы (рис. 1.5).

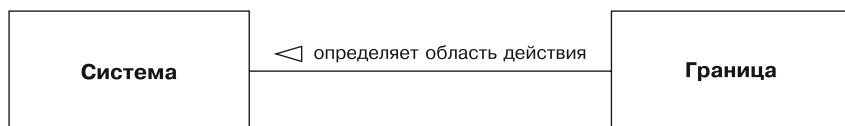


Рис. 1.5. Граница как определитель области действия системы

Существует множество типов границ, например следующие.

- **Физическая граница.** Это может быть некое ограждение вокруг системы, отделяющее ее от внешнего мира, или корпус, в котором размещены ее элементы. Примеры: корпус автомобиля, забор вокруг земельного участка, стены и двери, служащие границами помещения, и т. д.

- **Концептуальная граница.** Это нефизическая граница, которую можно представить, но не обязательно можно видеть. Пример: граница между автомобилем и спутником GPS, с которым он взаимодействует. Что будет границей системы в данном случае? Это передатчик и приемник в автомобиле, передатчик и приемник на спутнике, волны, которые передаются, или протоколы, которые используются как часть передачи?
- **Граница стейкхолдеров.** Разные заинтересованные стороны могут видеть систему по-разному, и местонахождение границы, по их мнению, может меняться в зависимости от стейкхолдера. Для примера возьмем двух разных стейкхолдеров для автомобиля. Пассажир может считать границей автомобиля его кузов, а автомеханик — концептуальную границу соединения между автомобилем и спутником.

Понятие «граница системы» позволяет усвоить несколько ключевых характеристик системы.

- **Что находится внутри границы.** Важно понимать, какие элементы системы считаются находящимися внутри ее границы, а какие — вне. Элементы, находящиеся внутри границы, помогают точно определить область действия системы.
- **Что находится вне границы.** Объекты, находящиеся вне пределов системы, считаются либо заинтересованными сторонами, либо обеспечивающими системами, а иногда (как мы выяснили несколько ранее), и теми и другими одновременно.
- **Где находятся ключевые интерфейсы.** Каждый раз при взаимодействии через границу системы формируется ее интерфейс. Выявление интерфейсов — важная часть системной инженерии, и именно границы позволяют выявить все точки взаимодействия системы с внешним миром.

С учетом этих факторов определить границы системы может быть не так просто, как кажется на первый взгляд, поскольку разные стейкхолдеры могут задавать разные границы. Это не всегда вызывает проблемы, но важно помнить об этом и убедиться, что эти различия не приводят к конфликтам.

Потребности

Каждая система должна иметь назначение, и оно задается набором *потребностей* (рис. 1.6).

Потребность описывает некое понятие, отражающее назначение системы.

Есть разные типы потребностей. На рис. 1.6 были показаны три из них.

- **Требование.** Это утверждение о том, что система должна (желательно) выполнять. Оно часто относится к конкретной функциональности системы.

Например, требование к автомобилю может заключаться в том, что водитель должен иметь возможность уменьшить скорость движения автомобиля с помощью педали тормоза, или что автомобиль должен иметь ремни безопасности, или что он должен развивать максимальную скорость не менее 150 км/ч.

- **Свойство.** Представляет собой потребность более высокого уровня, которая не обязательно связана с конкретным свойством, но может относиться к набору свойств. Пример: автомобиль должен иметь адаптивный круиз-контроль, должен уметь самостоятельно парковаться или иметь систему предотвращения столкновений.
- **Задача.** Это потребность очень высокого уровня, имеющая отношение к системе в целом. Пример: автомобиль должен перемещать водителя и трех пассажиров на расстояние 500 км без дозаправки.

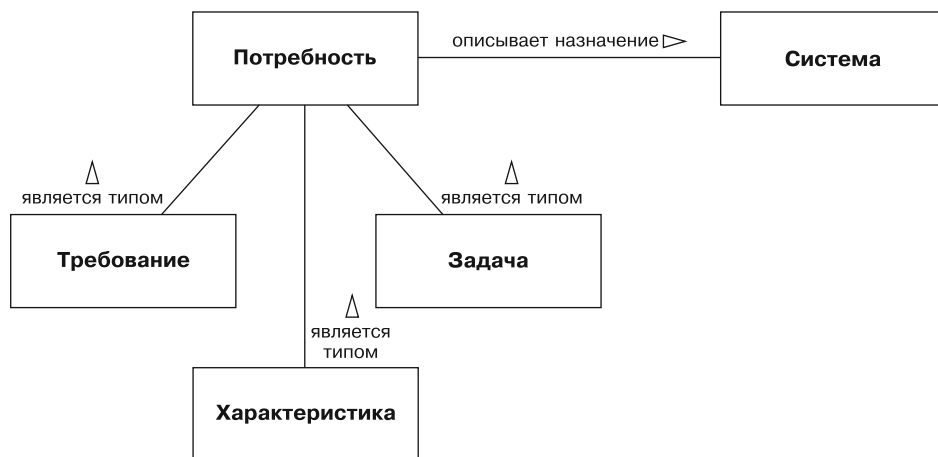


Рис. 1.6. Определение назначения системы

Необходимо подчеркнуть, что в разных отраслях и организациях существует много разных терминов для разных видов потребностей. Например, термин «способность» часто применяется в аэрокосмической и оборонной промышленности, а термин «свойство» более типичен для областей, связанных с транспортом, таких как автомобилестроение и железнодорожная отрасль. В некотором смысле нет разницы, какая терминология применяется, если это делается последовательно.

Ограничения

Любая система содержит *ограничения*, определяющие варианты реализации системы (рис. 1.7).

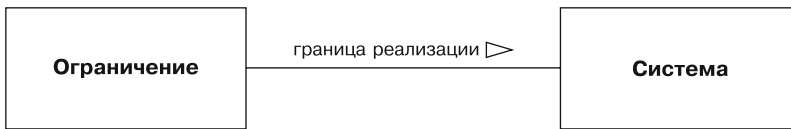


Рис. 1.7. Ограничения реализации системы

Нередко ограничения сгруппированы в несколько категорий.

- **Ограничения качества.** Почти во всех системах будут ограничения, связанные с источниками лучших практик, такими как стандарты. Обычно определяется ряд стандартов, которым должен соответствовать подход, применяемый при создании системы. Эти стандарты обычно относятся к процессам разработки, которые используются для описания общего подхода к системной инженерии. Например, стандарт, который часто применяется в автомобильной промышленности для создания автомобилей, — это ISO 26262.
- **Ограничения реализации.** Будут определять границы способа создания системы. Пример: ограничение использования материалов (алюминий, а не сталь) при производстве автомобиля.
- **Ограничения среды.** Все системы где-то разворачиваются, и многие будут существовать в естественной среде, что может привести к некоторым ограничениям. Пример: ограничения для автомобиля по выбросам в окружающую среду для снижения вредного влияния на нее.
- **Ограничения безопасности.** Почти на все системы накладываются ограничения, гарантирующие, что система может работать безопасно, особенно когда что-то в ней выходит из строя. Например, автомобиль может иметь функции, которые призваны защитить водителя и пассажиров при аварии.

Этот список далеко не полон.

Кроме того, важно помнить: ограничения могут быть сложными и относиться сразу к нескольким категориям. Например, для автомобиля может устанавливаться ограничение, согласно которому все материалы, из которых он состоит, можно подвергать вторичной переработке. Это ограничение относится к двум категориям: ограничения среды и ограничения реализации.

Еще один немаловажный момент: некоторые из ограничений относятся к разным стадиям жизненного цикла системы. Жизненный цикл — важная концепция, и о ней мы еще поговорим позже.

Ограничения также нередко описываются как особые типы потребностей, поскольку часто относятся к конкретным потребностям, а не к самой системе. Подробнее эту тему мы будем обсуждать в главе 6, посвященной именно потребностям.

Краткие итоги

Все концепции, которые мы обсуждали выше, можно объединить, чтобы увидеть, как они соотносятся с понятием системы (рис. 1.8).

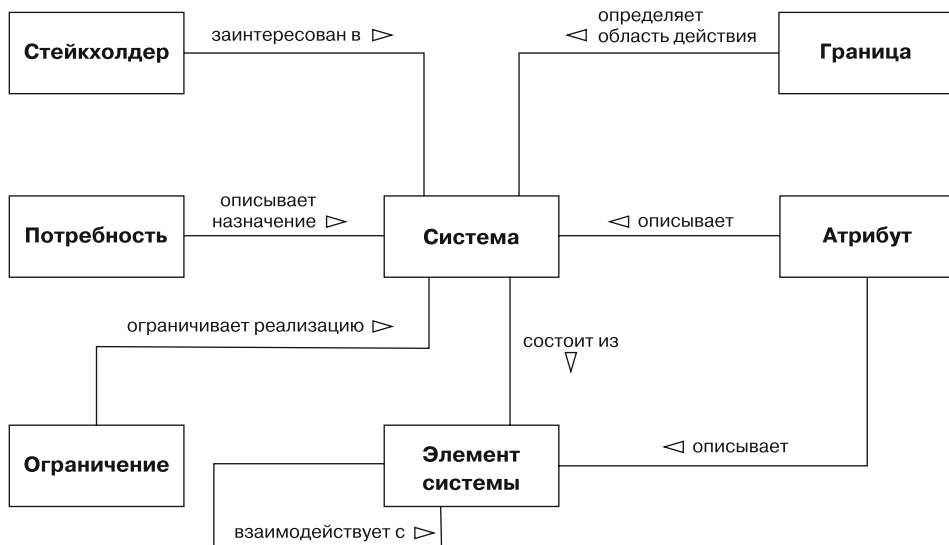


Рис. 1.8. Основные концепции, связанные с системой

Все эти концепции важно понимать, поскольку они будут использоваться далее.

Определение системной инженерии

Существует множество определений термина «системная инженерия»; кроме того, в разных публикациях эти определения обсуждаются, сравниваются и противопоставляются (Holt, Perry, 2019). В этой книге будет использоваться базовое определение из стандарта ISO 15288 (ISO, 2015) (в свою очередь, оно используется в книге *INCOSE Systems Engineering Handbook* «Руководство INCOSE по системной инженерии» (INCOSE, 2016)). В ISO 15288 термин «системная инженерия» определяется как

реализация успешно работающих систем.

Это определение схематически показано на рис. 1.9.

Схема может показаться тривиальной, но позволит соотнести общий термин с остальными концепциями, которые мы будем обсуждать далее в главе.

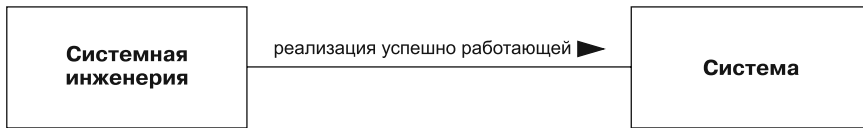


Рис. 1.9. Базовое определение системной инженерии

Определение простое, но полезное. В то же время при его прочтении необходимо учитывать несколько факторов.

- Системная инженерия — это междисциплинарный подход, принимающий во внимание все области инженерии: механическую, электрическую, гражданскую, программную и т. д. Тем не менее важно понять: она не сводится только к инженерным дисциплинам, а охватывает и многие другие области, такие как управление, математика, физика, психология — и это не полный список.
- Системная инженерия применяется на протяжении всего жизненного цикла системы и не ограничивается только одной какой-то стадией. Это значит, что она принимается во внимание сразу же с момента, когда появляется идея системы, и вплоть до вывода этой системы из эксплуатации. Даже при работе только над одной стадией жизненного цикла важно помнить об остальных стадиях.
- Системная инженерия не отменяет необходимость думать своей головой (системные инженеры не должны слепо следовать указаниям), и для ее эффективности требуется изрядная доля здравого смысла.

Учитывая эти соображения, первоначальное определение можно расширить так (Holt, Perry, 2008):

Системная инженерия — это основанный на здравом смысле междисциплинарный подход, который обеспечивает реализацию успешно работающих систем.

Теперь, когда определения инженерии даны, необходимо понять, для чего она нужна.

Почему необходима системная инженерия

Потребность в системной инженерии объясняется просто. В реальной жизни что-то может пойти не так. Проекты срываются, самолеты падают, наносится ущерб окружающей среде, люди получают травмы или гибнут, программное обеспечение и ИТ-системы парализуют работу организаций, а целые общества страдают из-за несогласованности в управлении и государственном регулировании — все это последствия системных сбоев на том или ином уровне.