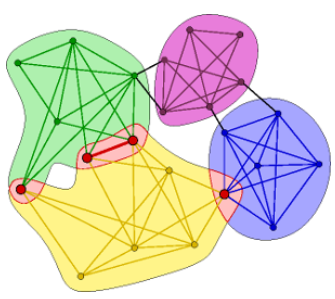
**Системное мышление** — это практический подход к восприятию мира, который значительно ускоряет способности анализировать, принимать решения и учиться. Практический потому, что сформирован практикой, а не вырос из абстрактных математических теорий.  
  
Если вам знакома аббревиатура [ТРИЗ](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B5%D1%82%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D1%85_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87), то я бы сказал, что методы ТРИЗ — это набор частных случаев применения системного мышления в физическом производстве.

Система — это абстрактное понятие, позволяющее нам структурировать окружающий мир в удобном для анализа виде.  
**Система — это совокупность связанных между собой ~~штук~~ объектов. По сути это связанные вещи, которые служат для чего то одного. Если какие то вещи служат для разного, то они не составляют систему единую (например автобусная остановка и двигатель машины). Автобусная остановка включает жилой квартал, людей и их требования. Служит для того что бы люди перебрались из точки А в точку Б. Люди составляют часть этой системы. Двигатель включает клапаны и прочие детали и не составляет часть системы с автобусной остановкой.**  
  
Связанных — в смысле хоть как-то влияющих друг на друга: передающих информацию, скреплённых сваркой, дёргающих друг дружку за верёвочки, etc.  
  
Важно понимать: только человек решает что назвать системой, природа не различает систем. По сути, любой набор любых сущностей можно обозвать системой, но это недейственный метод. Система должна быть концептуально целостной, только тогда её использование будет полезным.

Если представить граф, вершинами которого будут все сущности в анализируемой нами области, а рёбрами — связи между ними, то хорошими кандидатами в системы станут кластера сильно связанных вершин.  
  
Выглядеть это может примерно вот так.



Любая система может состоять из подсистем и быть частью одной или нескольких метасистем.

Например:

* двигатель — система из клапанов и прочих деталей
* автомобиль — система из различных устройств, одним из которых является двигатель (подсистема)
* автомобильная дорога — система из инженерных сооружений, транспортных средств и пешеходов (подсистемы)
* автобусная остановка — система, являющаяся частью метасистем «автомобильная дорога» и «жилой квартал»

Подсистема какой то системы является для самой системы – объектом

Например двигатель, двери и тд для автомобиля являются объектами. Хотя когда мы тестируем двигатель, то у него есть объекты клапаны, и тд.

Когда мы тестируем систему, то нет смысла тестировать объекты их подсистем, достаточно только протестировать сами эти подсистемы (которые как описано выше – являются объектами самой системы), а так же не связанные системы между собой системы.

Например при тестировании автобусной остановки, нам не важно как работает двигатель у машины, нам скорее важно расписание автобусов и успеют ли все люди доехать до работы. Или при тестировании двигателя нам не важно успевает ли человек на работу, нам важны именно значения системы двигатель

Если мы тестируем автомобильную дорогу, то проверяем ее ширину, протяженность, материал. Сколько машин на ней может одновременно ехать и тд. При этом нам не важны объекты самого автомобиля

Если мы тестируем автомобиль, то проверяем его габариты, что установлен двигатель, двери и тд, при этом нам не важно как работает и работает ли вообще двигатель

Если же мы проверяем двигатель, то проверяем что он может привести в движение машину

Если переложить это на Software Testing, то например есть функциональность указания Email в настройках профиля. Мы ожидаем что после нам на Email придет сообщение об успешно изменении Email на другой. А так же на новый указанный Email придет так же оповещение. Здесь в эту систему включается следующие объекты:

* Старый Email
* Новый Email
* Настройки пользователя

При этом указание Email в настройках пользователя является подсистемой другой системы – Настройки. И при тестировании подсистемы Email нам не нужно тратить время на проверку других подсистем, так как они не связаны – например указание Email и добавление аватарки в настройках. Тем более нам не нужно тестировать не связанные системы - например указание Email и добавление товара в корзину.

**Что нужно проверять, так это объекты тестируемой подсистемы/системы, а так же связанные подсистемы/системы**.

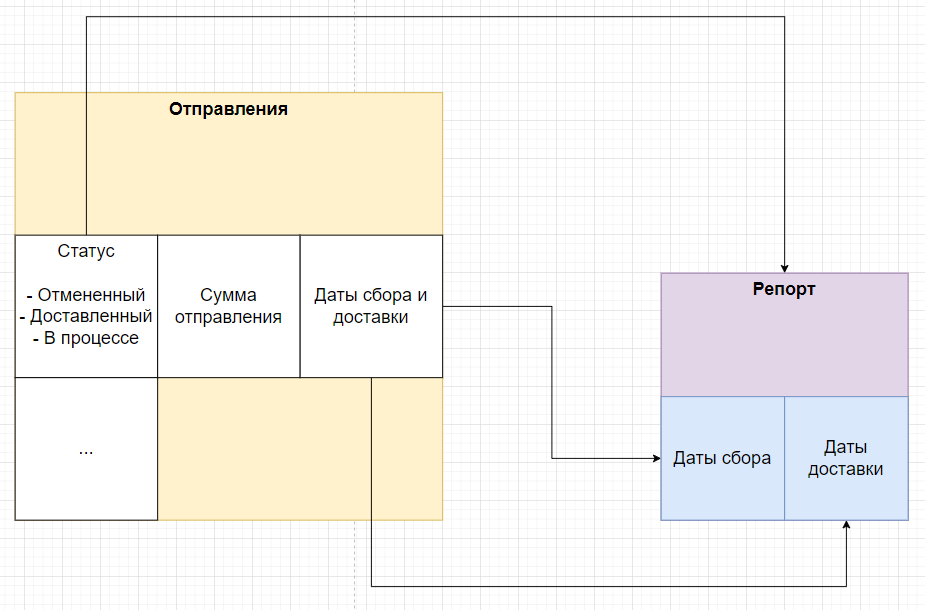
Под объектами я понимаю Старый Email, Новый Email, Настройки пользователя

Под связанными системами я понимаю: при покупке товара приходит уведомление на Email. В таком случае, мы можем сменить Email и посмотреть придет ли оповещение о покупке на новый Email. Можно взять за практику, если какая та функциональность (система/подсистема) – (покупка товара), описывает взаимодействие с другой функциональностью, которая будет представлена объектом какой то системы/подсистемы –(Email). То протестировать реализуемую функциональность вместе со связанный объектом и его системами (изменение Email, удаление аккаунта (так как Email привязан к аккаунту))

**Например**: добавили возможность делать репорт – какие отправления делал пользователь за выбранную дату

Объекты:

* **Даты указываемые при генерации репорта** 
  + Дата сбора
  + Дата получения
* **Отправления**. У отправления есть куча атрибутов (Статус отправления – Отмененный, Доставленный, В процессе ; Сумма отправления ; Даты сбора и доставки ; …). Нам нужны именно те атрибуты, которые связаны с задачей. Так как задача про даты, то и атрибуты выбираем даты. Так же отмененные по условию не должны попадать
  + Дата сбора
  + Дата получения
    - Так же думаем можно ли поменять как то атрибуты Дат сбора и доставки
  + Статус отправления – Отмененный
    - Так же думаем можно ли как то поменять атрибут статус



При тестировании, соответственно, проверить изменение статуса, даты сбора и получения. Посмотреть что будет выводиться в отчете