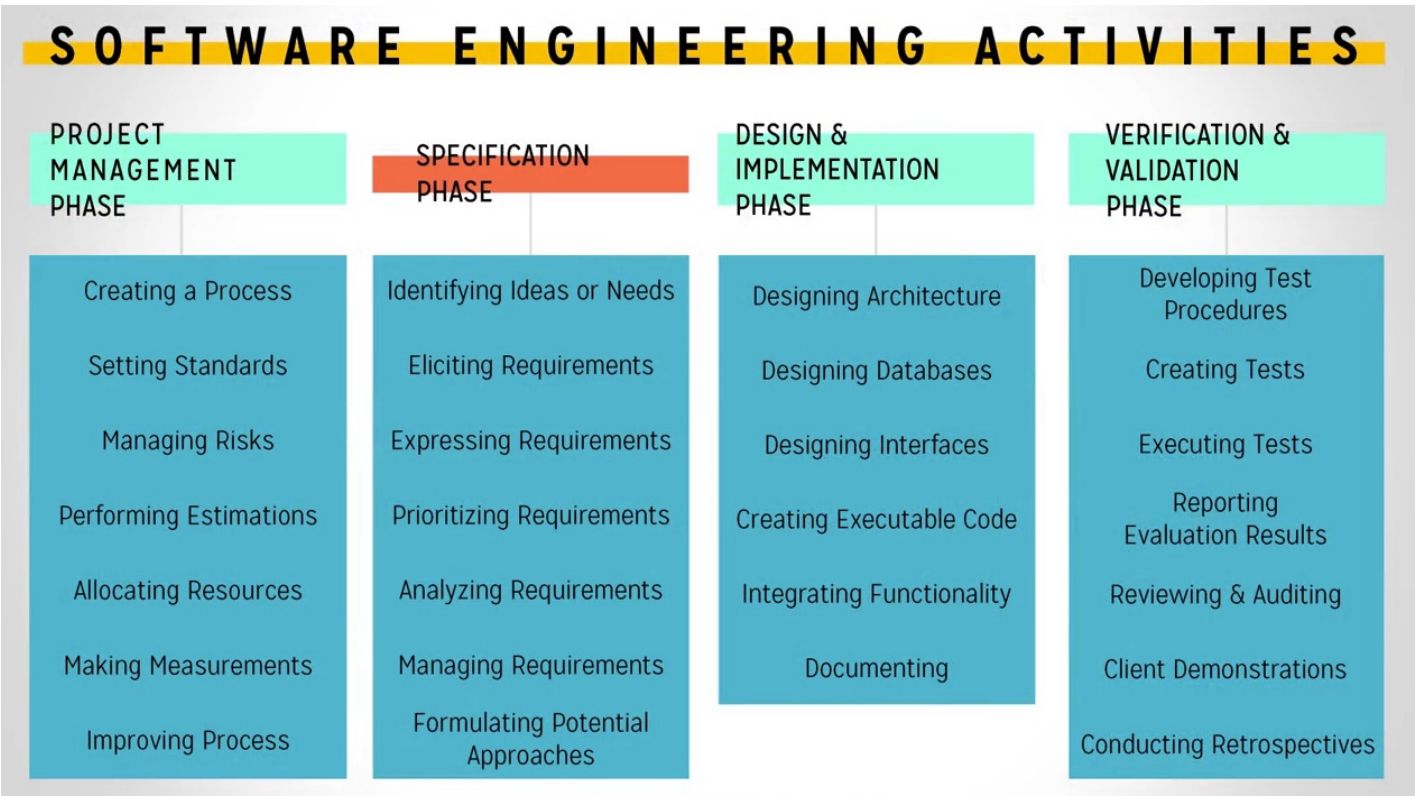
client needs and Software requirements

⚠ Requirements

↳ specific descriptions of your client's needs



⚠ Product done right ≠ right product

⚠ Elicitating requirements ≠ gathering requirements

has more investigation

⚠ Business requirements outline purpose of a software

⚠ Business rules constrain how the product functions

⚠ Use cases are the good way to show the relationship between the system and the user

⚠ functional requirement

↳ a behaviour that the product should do or support

⚠ non-functional requirements

↳ serve as a description of "how well" a product must perform

↳ types

- ↳ accuracy
- ↳ dependability
- ↳ security
- ↳ usability
- ↳ efficiency
- ↳ performance
- ↳ maintainability

⚠ External interface requirements

↳ outline where the product is situated (how connects or communicates)

⚠ physical setting requirements

↳ describe how the product should be designed

⚠ Vision

↳ the long-term strategic concept of the ultimate purpose and form of a new system

⚠ vision vs. scope

Vision	Scope
encompasses what the product will eventually do to satisfy a user's need	encompasses what can be realistically achieved within the current project

⚠ How to defend against scope creep

- ↳ Have the client prioritize requirements
- ↳ make expectations clear
- ↳ Draw the scope with your client
- ↳ Ask "Is this in scope?"
- ↳ Evaluating the impact of proposed changes

⚠ requirements → what / design → How

⚠ end user → who's this product for

- types
  - primary, secondary, and teritiary

⚠ types of user's limitations

- perceptual limitations (ex: color blindness)
- physical limitations (ex: left-handed people)
- cognitive limitations
- cultural limitations

⚠ ways of elicitating user's requirements

- interview end-users
- conduct feasibility studies
- watch end-users using the product
- consult user manuals for similar products
- create a glossary for the product

⚠ Use "Asking why" technique to understand the client's requirements better

⚠ Use case

↳ a way to identify, clarify, and organize the details of a task

⚠ Use case consists of:

Name	
Participating Actors	
Goals	
Triggers	
Pre-Condition	
Post-Condition	
Basic Flow	
Alternate Flows	
Exceptions	
Qualities	

⚠ Use case Diagram

↳ high level visual representation of all the tasks supported by the product

⚠ wireframe

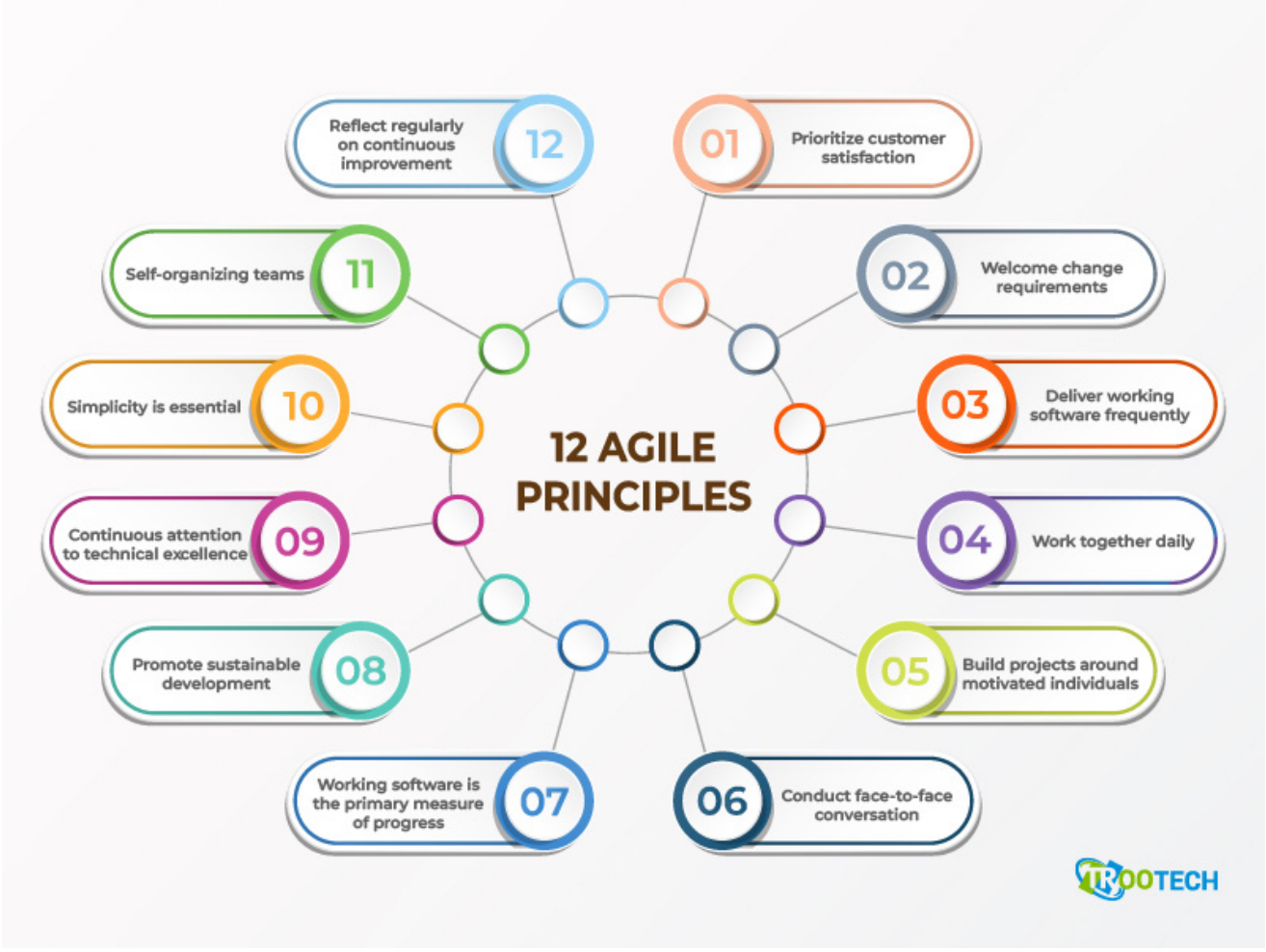
↳ a basic visual representation of a product. Another term for a wireframe is a mock up

⚠ Story board

↳ a sequential, visual representation of an interaction

↳ shows the communication & interaction of software pages

⚠ Agile Principles



⚠ A user story is a simple way of expressing requirements

↳ format { as a _____ ,
I want _____ ,
so that _____ .

⚠ A good user story is one which clearly outlines a specific software requirement in the product

↳ INVEST → Independable
- Negotiable
- Valuable
- Estimable
- Small
- Testable

(should be)

⚠ An "Epic" is a user story that is nearly insurmountable. They're vague, broad descriptions of a requirement

⚠ An "Acceptance Test" is a check for whether a requirement is met

⚠ Acceptance Criteria are specific conditions which must be met

⚠ A product backlog is a list of software features which you and your team intend to develop

↳ isn't just limited to user stories

⚠ In Scrum, Product owner is responsible for creating and prioritizing user stories

⚠ Story Maps are a great way of representing your product backlog with more visual detail

- gives perspective how user stories relate

⚠ High quality user stories should be

- correct
- complete
- clear
- consistent
- feasible
- traceable

⚠ Ambiguous Requirement Categories

- indirect words
- vague words
- persuasion words
- completion words
- qualifiers
- comparatives
- quantities
- pronouns
- positional words
- temporal words
- joining words

⚠ indirect words ~ may, will, ---

⚠ vague words ~ processed, operated, handled, ---

⚠ completion words ~ and so on, ---

⚠ persuation words ~ clearly, obviously, ---

⚠ qualifiers ~ all, none, often, only, ---

⚠ comparatives ~ Car A is bigger than Car B

⚠ quantities ~ some, most, few, ---

⚠ Pronouns ~ them, him, it, ---

⚠ positional words ~ after, before, following ---

⚠ temporal words ~ when, for, untill, ---

⚠ joining words ~ and, or, both, ---