

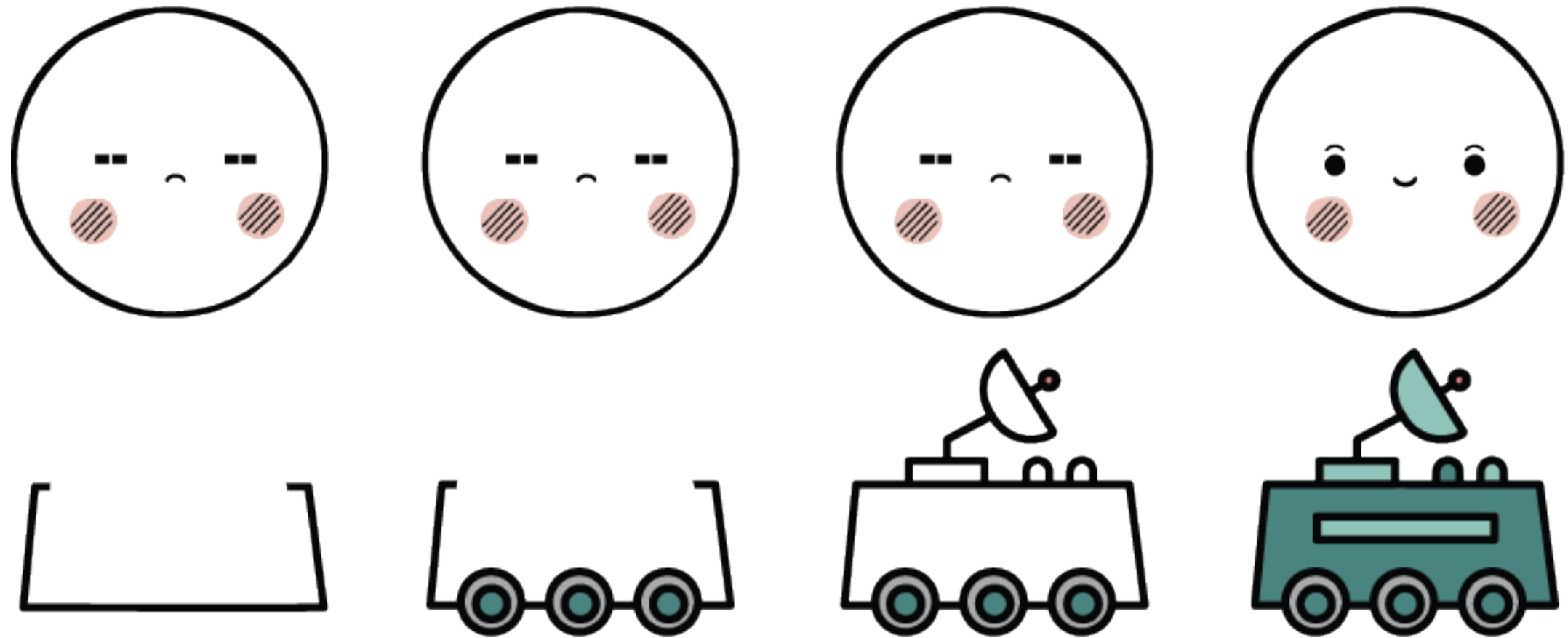


AGILE 101

Modern software development practices

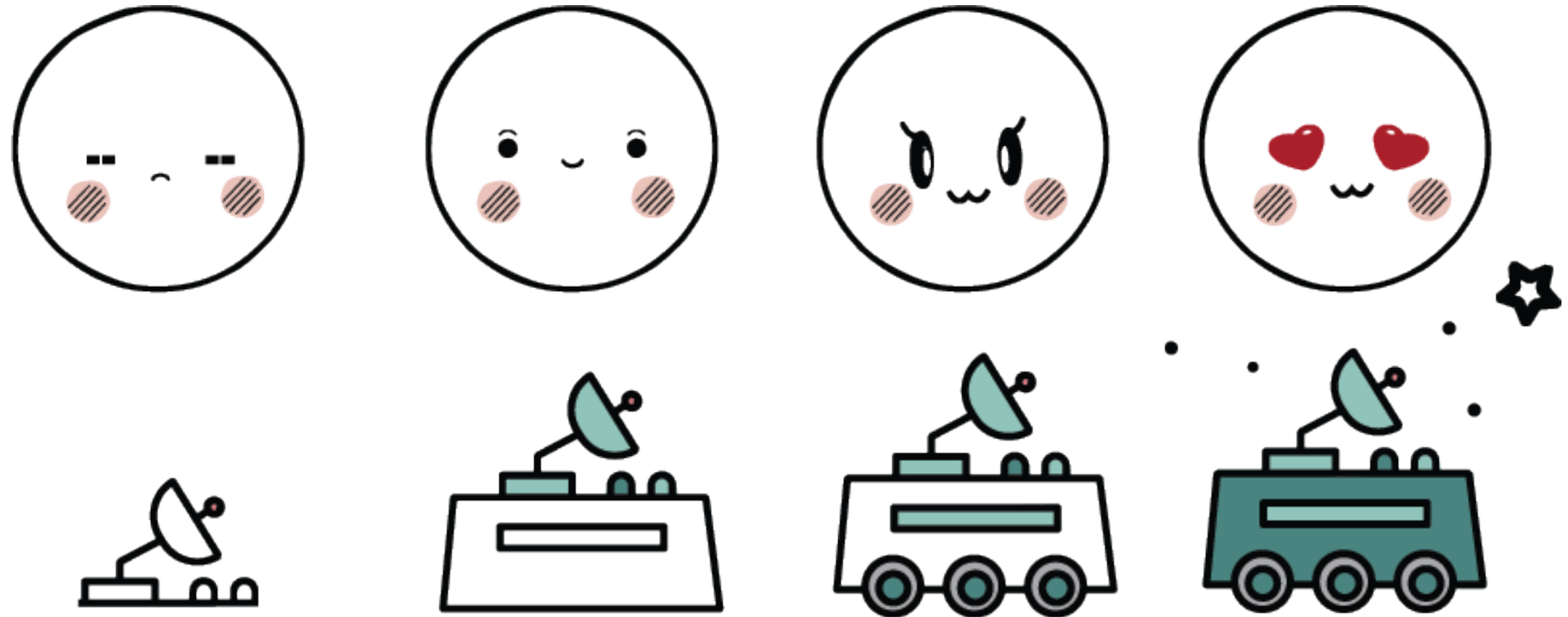


“WATERFALL” DEVELOPMENT



Big Plans at the beginning, no value until very end.

“AGILE” DEVELOPMENT



Creating value incrementally, able to change with circumstances

WATERFALL VS AGILE DEVELOPMENT

- Follow the plan developed at the beginning of the project
- A then B then C then D
- Continue on even if it's dumb
- Causes failures if anything changes during the project
- Only the entire, completed product is delivered at the end
- Value must wait for completion
- Allows for progressive changes, alterations to original plan
- Cyclical - "eats the elephant" one bite at a time
- Able to change with the market, technology, new ideas
- Valuable pieces of project are delivered whole every sprint
- Continually delivering value

WATERFALL



Waterfall requires detailed planning at the beginning of a project.



All the steps are laid out, dependencies mapped, and you move to the next stage only after completing the previous one.



Pros.



Best for projects that deal with physical objects – from a construction project to a hardware installation project.



Best for projects with defined tasks and phases that must be completed in a specific sequence (e.g., build the first floor of a building before the second floor).



Project plans are repeatable for identical or similar projects in the future.



Cons.



Requires substantial scope and schedule planning before work begins.



Scope changes can be slow and require formal change control processes.



Less effective for software, design and other non-physical or services-based projects.

AGILE



This is a fast and flexible approach to project management based on principles of collaboration, adaptability and continuous improvement.



Unlike the orderly stages of a waterfall approach, agile project management is typically set up in quick, iterative project release cycles.



Pros.



Best for projects that deal with services-oriented and non-physical deliverables like code, copywriting and design projects.



Allows for quick course correction based on stakeholder feedback.



Empowers project teams to work creatively and efficiently.



Includes engagement and collaboration from all team members.



Cons.



Not suited for projects with strictly defined requirements and scope.

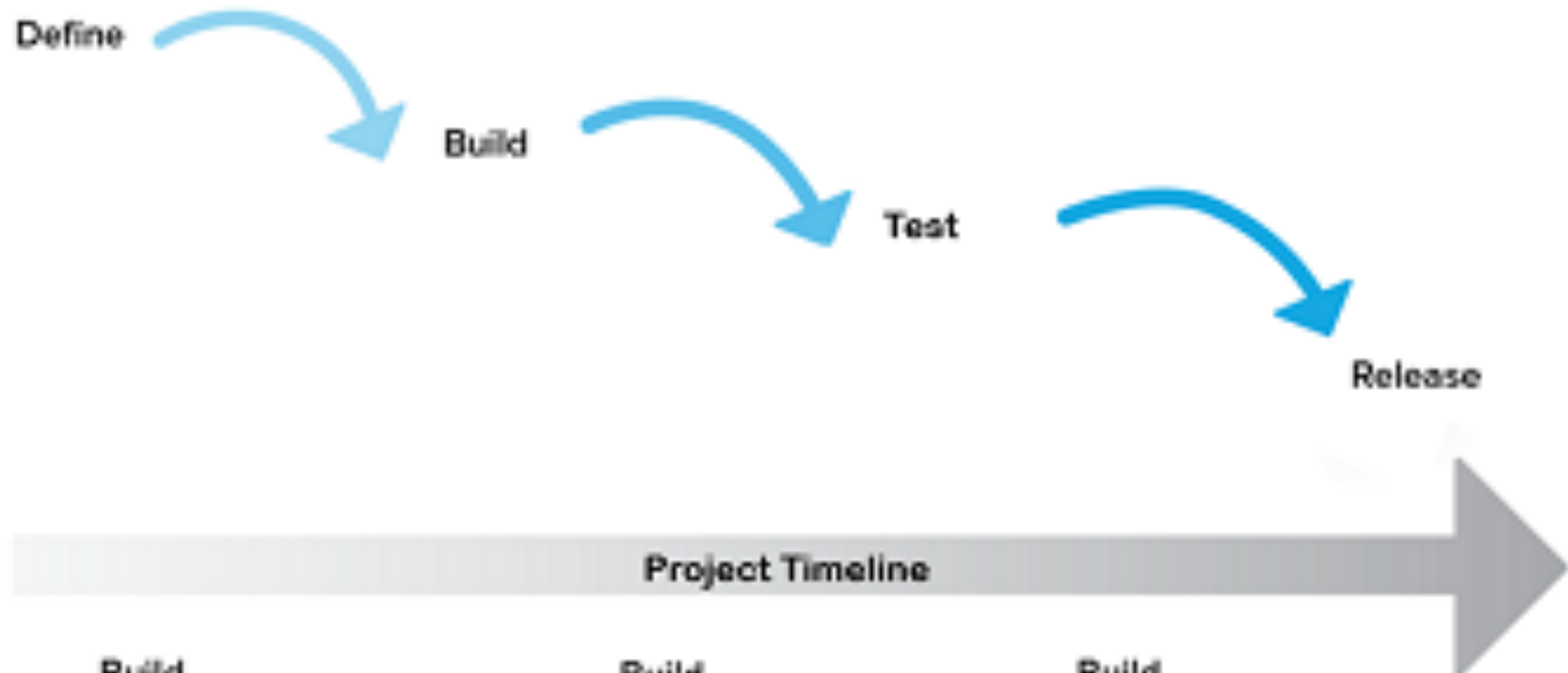


Uncertainty around scope and schedules can make stakeholders and executives nervous (at first).



Requires vigilant backlog and documentation maintenance, and tech debt management.

WATERFALL



AGILE

RULES OF THE GAME: AGILE LEGO SIMULATION

- **Vision:** Build the best and most beautiful plane in the market using Lego blocks.
- **Facilitator role:** I am the client.
- **Business value:** Each story has a business value, that means how important is the story for the client and to success in their business. The client has provided the business value as requested.
- **Definition of Done:** Nothing is accepted unless the client has accepted it.
- **Goal:** To maximize the Return on Investment of the client. That is to maximize the business value per effort applied in each sprint.

RULES OF THE GAME: AGILE LEGO SIMULATION

- 7 or 8 sprints of 5 minutes each
- At the end of each sprint we will do a sprint review and you will demo your product
- I will provide feedback on which items I like and do not like (according to my own acceptance criteria)
- You CAN ask questions during gameplay

PREPARATION

- Select roles, divide responsibilities
- Every person must have a distinct role
- No two people will perform the same role on a team
- Name your team
- I will give you my Requirements - your team is responsible for noting these and delivering.

SPRINT 1

- We're building an airplane!

RETROSPECTIVE:

- What should we **STOP** doing?
- What should we **START** doing?
- What should we **CONTINUE** doing?
- Are there any **RISKS** to the completion of this project?