



Introduction to IT project management

KRYSTIAN WOJTKIEWICZ, PHD

Software Development Life Cycle

SDLC



Planning

Project conditions assessment

Costs calculation

- Workload
- Resources

Scheduling

Providing feedback to stakeholders

Planning

Scope

Aim

The course

Boundaries

Requirements definition



considered an element of planning

define what the software should do

allow to focus on the important parts of the software

allow to identify the necessary resources

Design and prototyping

Architecture



GUI



Platform



Technology stack



Communication



Security



Software development

Proper software development

Depending on the project, they are carried out by teams of different sizes

Recommended use of supporting software

- Access control
- Code management

Value instructions and explanations

- User's Guides
- Comments in the code

Testing



Non-functional Functional



- Efficiency
- Load
- Utility
- Reliability



Deployment

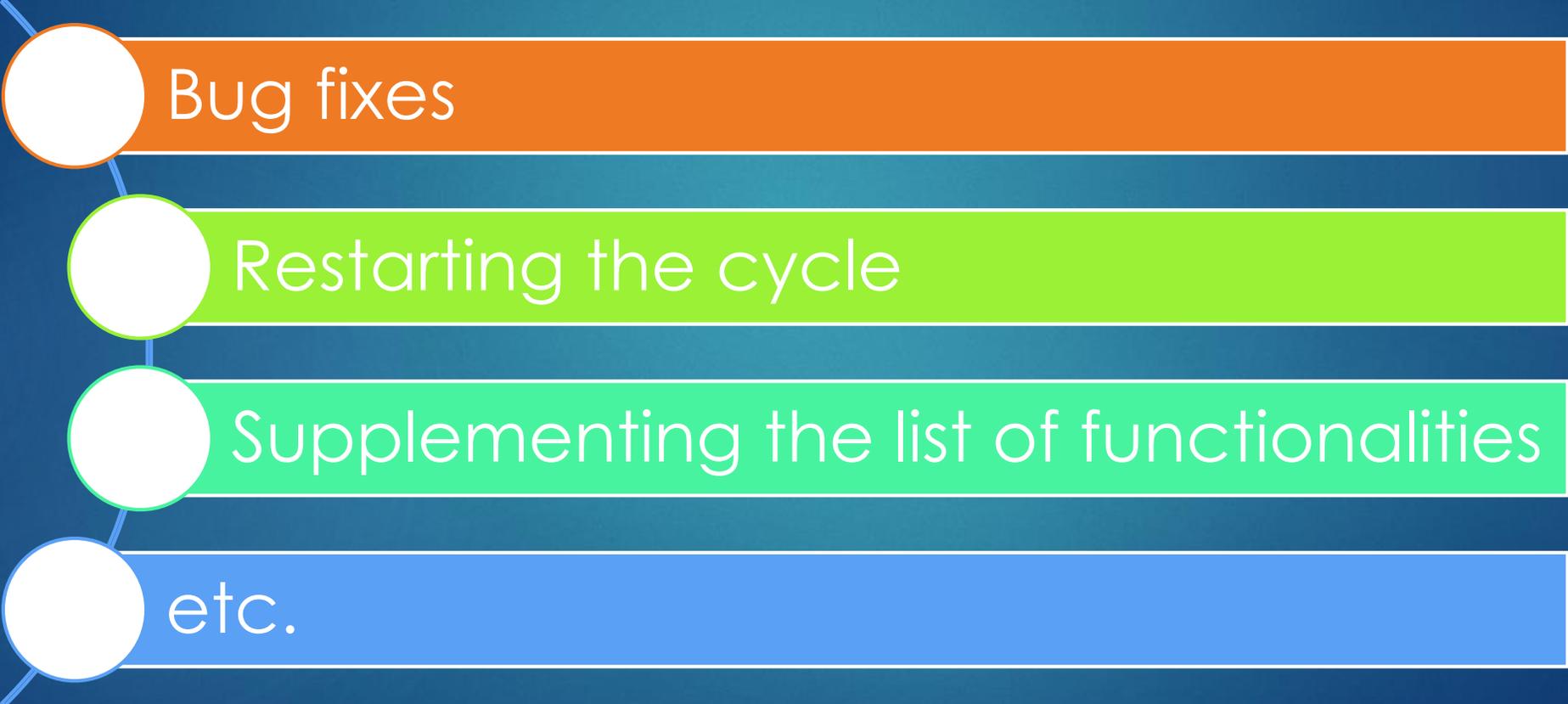
Automatic

For simple systems
For subsystems and modules

Complex

System change
Launching a new, unknown service

Operation and maintenance



Bug fixes

Restarting the cycle

Supplementing the list of functionalities

etc.



How the customer explained it



How the Project Leader understood it



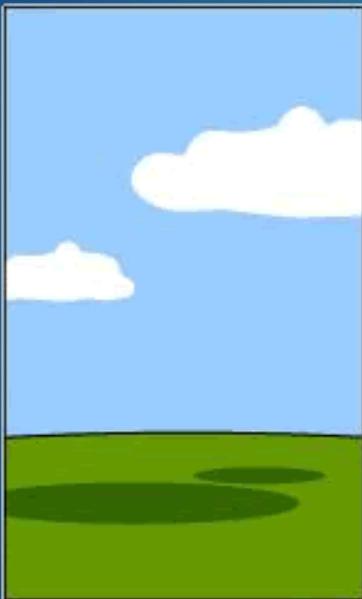
How the Analyst designed it



How the Programmer wrote it



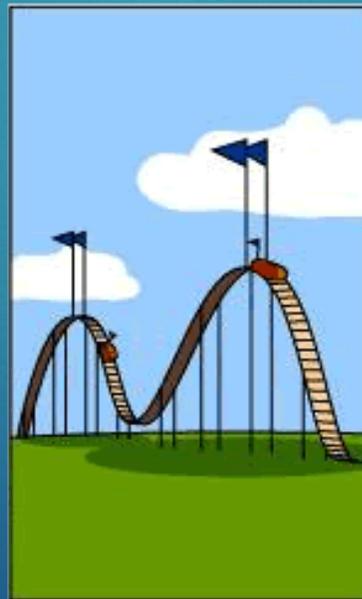
How the Business Consultant described it



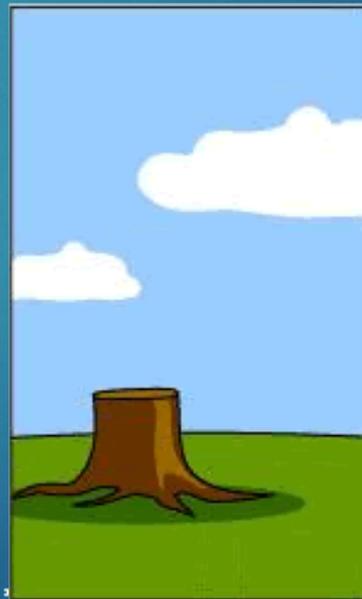
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

Methodology vs Framework

project management methodology

Project management methodology offers a clear project roadmap that lists all the steps required to deliver a project successfully.

project management framework

The project management framework provides structure and direction to a project. Frameworks guide projects to their goal while being flexible enough to adapt to evolving conditions.

Framework

Gives an overview of how guidelines can be implemented

Offers space for creative adaptation

Preferred by experts

Makes it hard to develop and implement performance metrics

Leaves room to include other practices and tools

Traditional project management (PMBOK) is a framework

Methodology

Offers rigid rules and practices for completing a project

Is pretty rigid and prescriptive

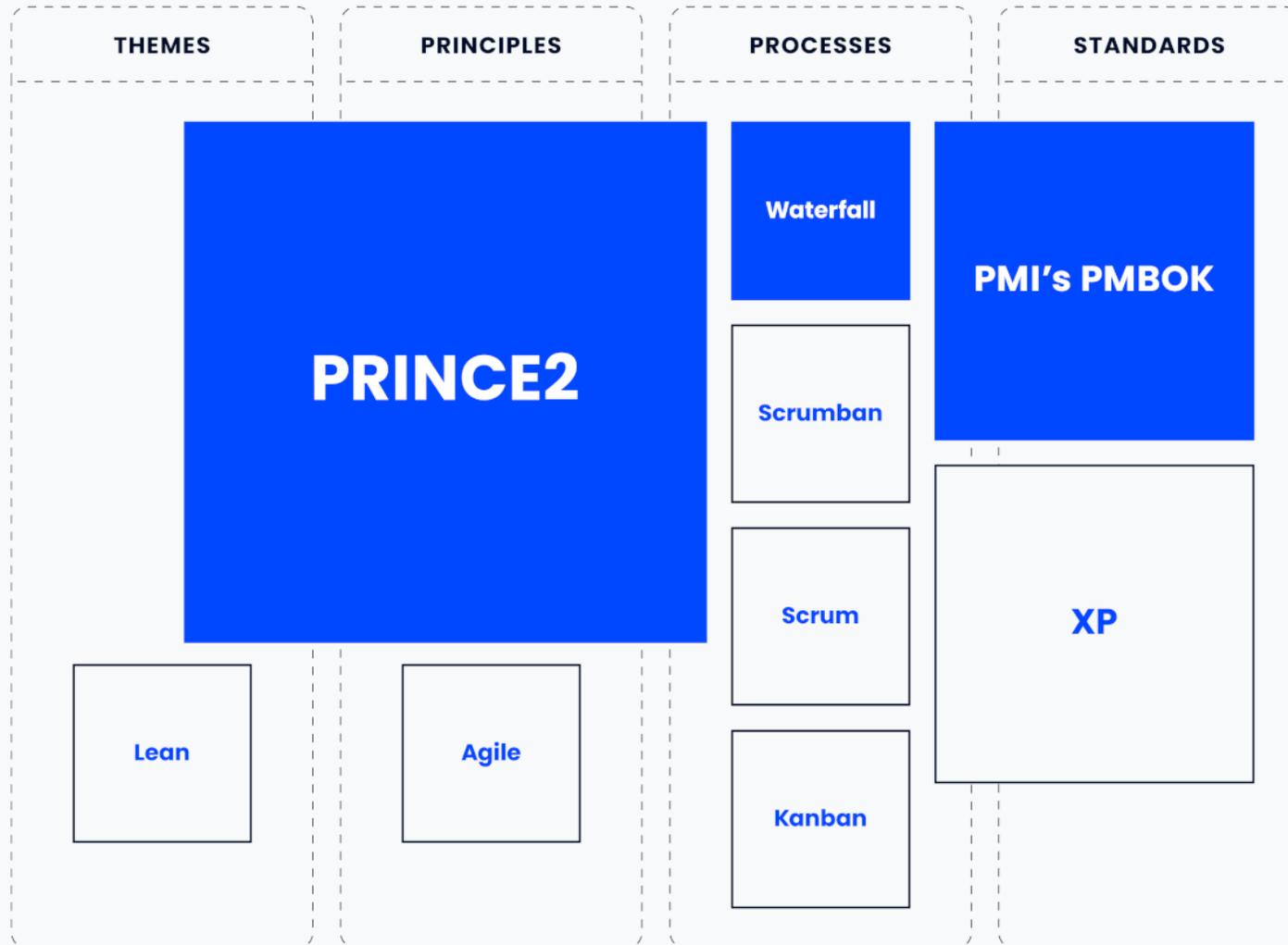
Preferred by beginners

Spells out all performance guidelines in granular detail

Cannot be embedded with other practices and tools

PRINCE2 is a well-known project management methodology

Project Management Methodologies



Increasing "method" specificity →

Themes

Principles

Processes

Standards

How to choose, what to choose?

Consider your project factors by their simplicity or complexity

Determine the rigidity or flexibility of your work environment

Consider what delivers the most value

Leverage your organizational goals

List your organizational and team values

Methodologies and frameworks

Agile

Collaborating to iteratively deliver whatever works

SCRUM

Enabling a small, cross functional, self-managing team to deliver fast

Kanban

improving speed and quality of delivery by increasing visibility of work in progress, and limiting multi-tasking

Scrumban

limiting work in progress like Kanban with a daily stand up like Scrum

Lean

streamlining and eliminating waste to deliver more with less

XP

Extreme Programming methodology– doing development robustly to ensure quality

Waterfall

planning projects fully, then executing through phases

PRINCE2

controlled project management that leaves nothing to chance

PMI's PMBOK

applying universal standards to waterfall project management

Agile

Values

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

Principles

1. Customer satisfaction through early and continuous software delivery
2. Accommodate changing requirements throughout the development process
3. Frequent delivery of working software
4. Collaboration between the business stakeholders and developers throughout the project
5. Support, trust, and motivate the people involved
6. Enable face-to-face interactions
7. Working software is the primary measure of progress
8. Agile processes to support a consistent development pace
9. Attention to technical detail and design enhances agility
10. Simplicity
11. Self-organizing teams encourage great architectures, requirements, and designs
12. Regular reflections on how to become more effective

Scrum

Scrum team roles

Product owner: Product expert who represents the stakeholders, and is the voice of the customer.

Development team: Group of professionals who deliver the product (developers, programmers, designers).

Scrum master: Organized servant-leader who ensures the understanding and execution of Scrum is followed.

Scrum events

Sprint: Iterative time boxes in which a goal is accomplished. Time frame does not exceed one calendar month and are consistent throughout the development process.

Sprint planning: Where the entire Scrum team get together—at the beginning of every Sprint—to plan the upcoming sprint.

Daily Scrum: 15 minute time boxed meeting held at the same time, every day of the Sprint, where the previous day's achievements are discussed, as well as the expectations for the following one.

Sprint review: An informal meeting held at the end of every Sprint where the Scrum team present their Increment to the stakeholders, and discuss feedback.

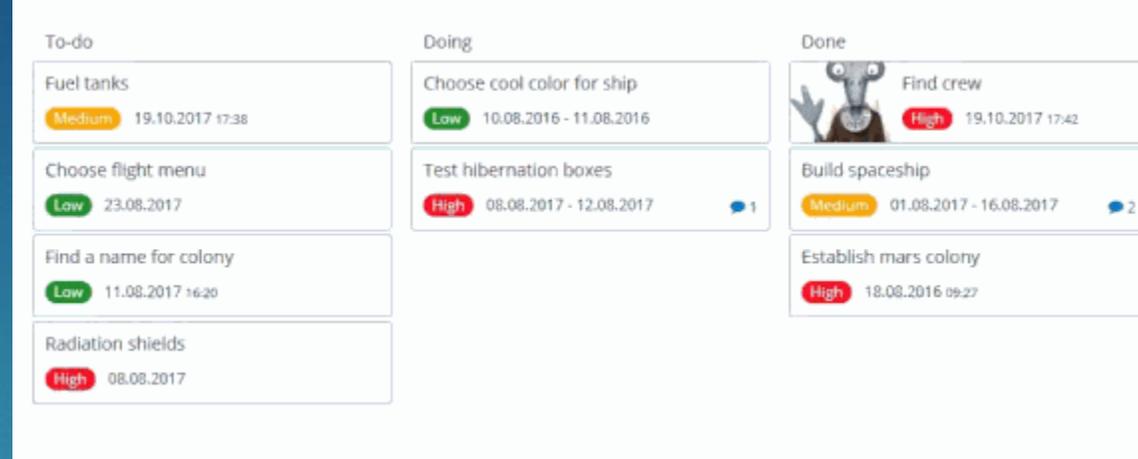
Sprint retrospective: A meeting where the Scrum team reflect on the proceedings of the previous Sprint and establish improvements for the next Sprint.

Scrum Artifacts

Product backlog: Managed by the Product Owner, it's where all the requirements needed for a viable product are listed in order of priority. Includes features, functions, requirements, enhancements, and fixes that authorize any changes to be made to the product in future releases.

Sprint backlog: A list of the tasks and requirements that need to be accomplished during the next Sprint. Sometimes accompanied by a Scrum task board, which is used to visualize the progress of the tasks in the current Sprint, and any changes that are made in a 'To Do, Doing, and Done' format.

Kanban



Kanban board

What's used to visualize the development process, a Kanban board can be either physical (a whiteboard, sticky notes, and markers) or digital.

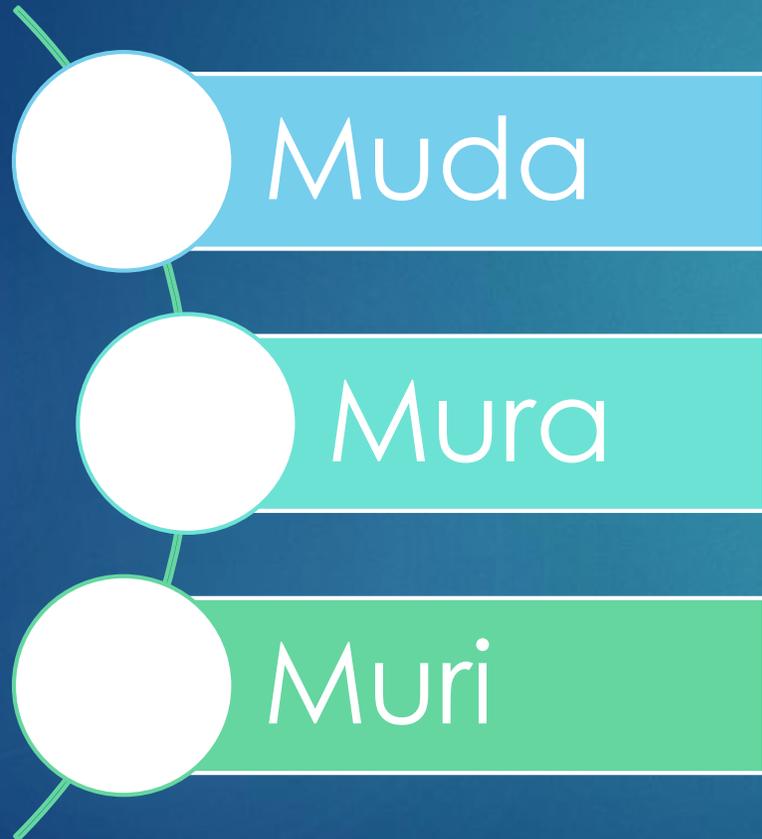
Kanban cards

Each Kanban card depicts a work item/task in the work process. Used to communicate progress with your team, it represents information such as status, cycle time, and impending deadlines.

Kanban swimlanes

Flowing horizontally, Kanban swimlanes are a visual element on the board that allows you to further distinguish tasks/items by categorizing them. Their purpose is to offer a better overview of the workflow.

Lean



Waterfall



WE ARE GOING AGILE...



Cascade Model

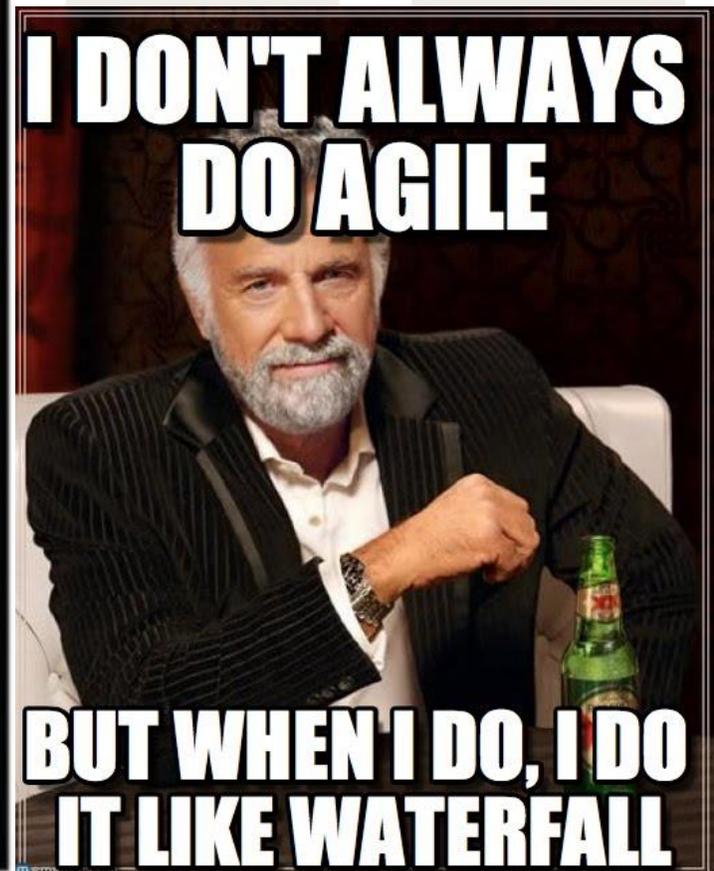
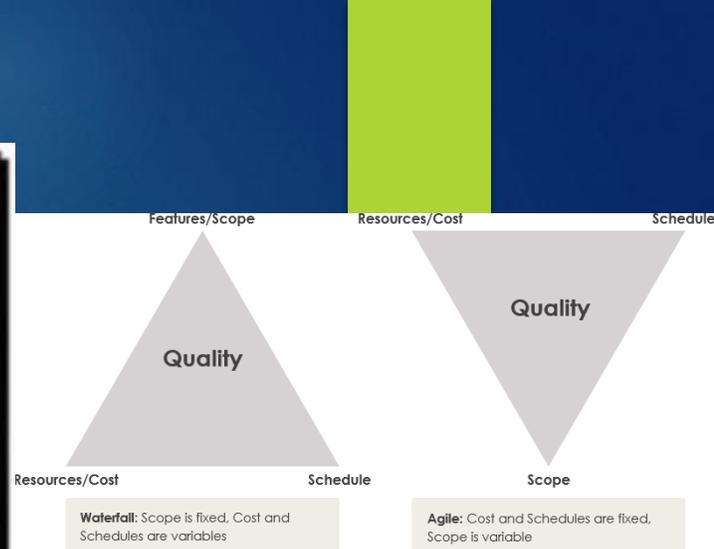
You want to go to Mars | You build a rocket | You test a rocket | You're on Mars!

Agile

You want to Mars, probably | You begin building a rocket | Now you want to Uranus | You're on the Moon

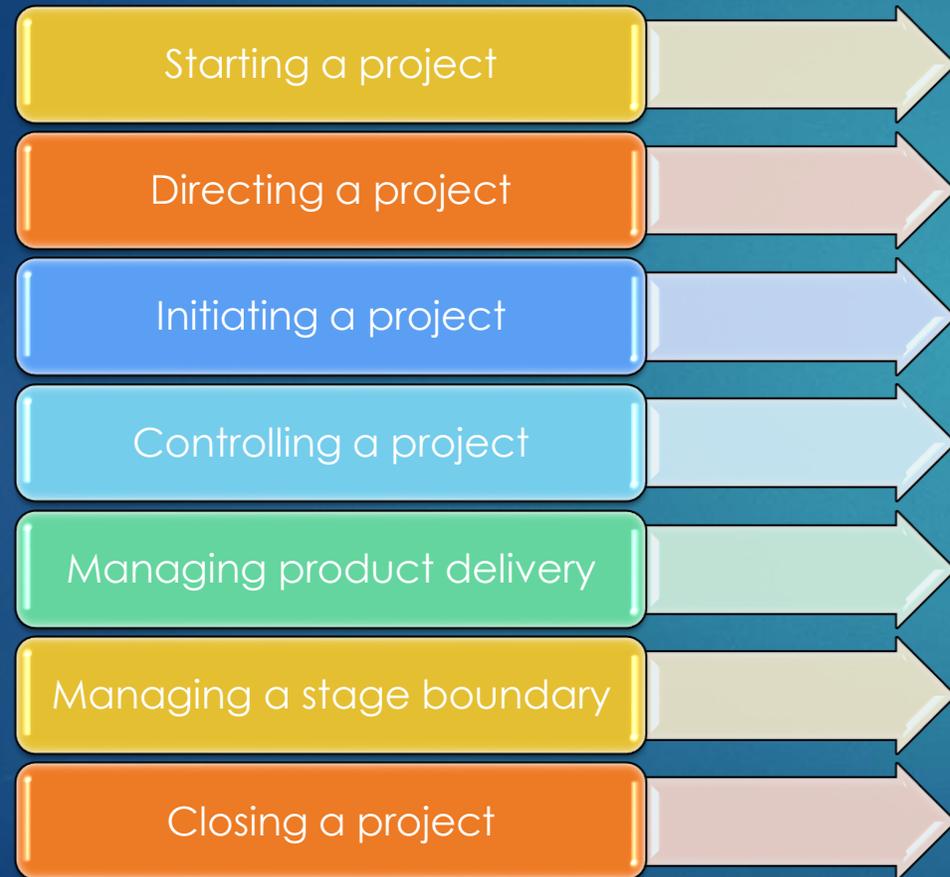
Kanban

You want to go to Mars | You break a task into a thousand small subtasks | A year later you are waiting for the armrests to be made



Prince2

(PProjects IN Controlled Environments)

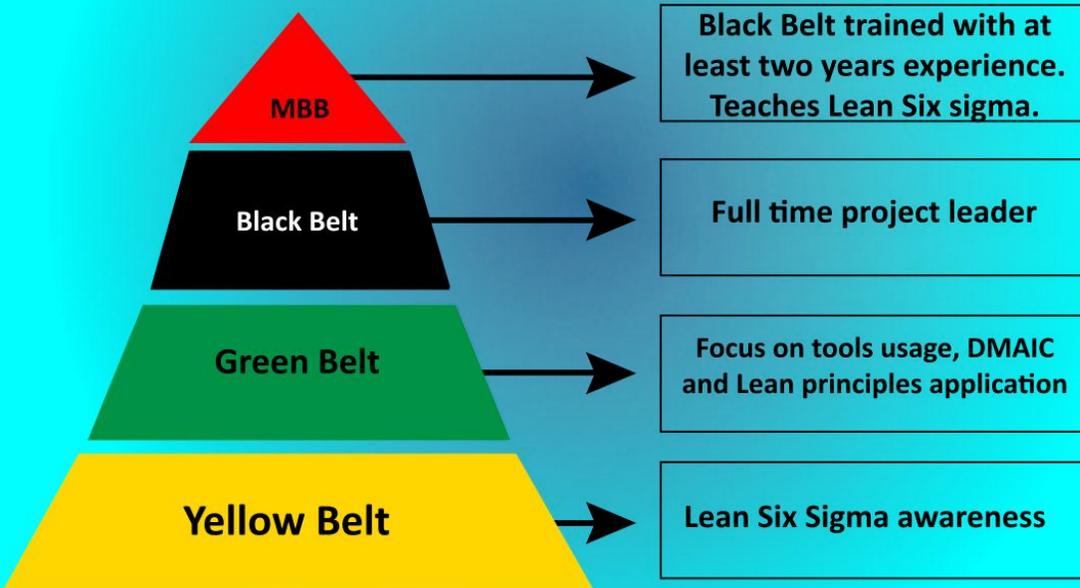


Six Sigma

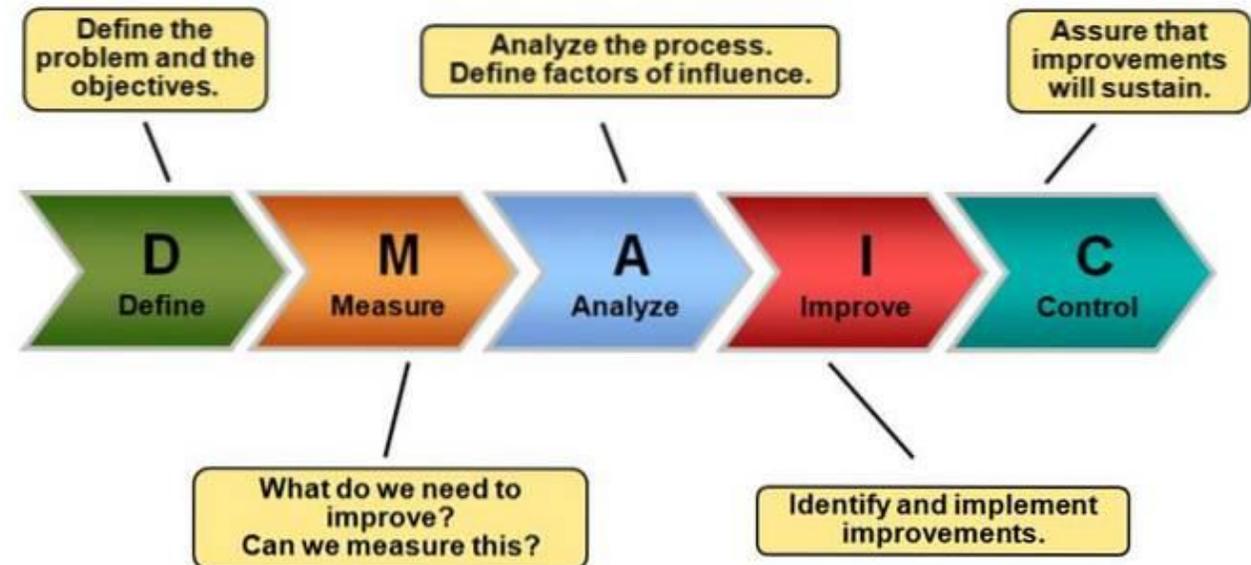
What is **Lean Six Sigma** ?



LEAN SIX SIGMA ORGANIZATION STRUCTURE



DMAIC Roadmap



PMI/PMBOK

Project
initiation

Project
planning

Project
executing

Project
performance

Project
closure

