Introduction to the course

6th Feb 2026

Nazih Errahel

erraheln@ex.istu.edu

OUTLINE

Survey

Software Development Life Cycle

Waterfall & agile model

DevOps Motivation

DevOps history & stats

What are Silos, culture Goals, adoption, values, benefits, etc.

DevOps Phases (Plan, code, build, test, ...)

Continuous everything culture in DevOps

Know audience

Quick Survey

https://www.menti.com
Code: 1142 8343

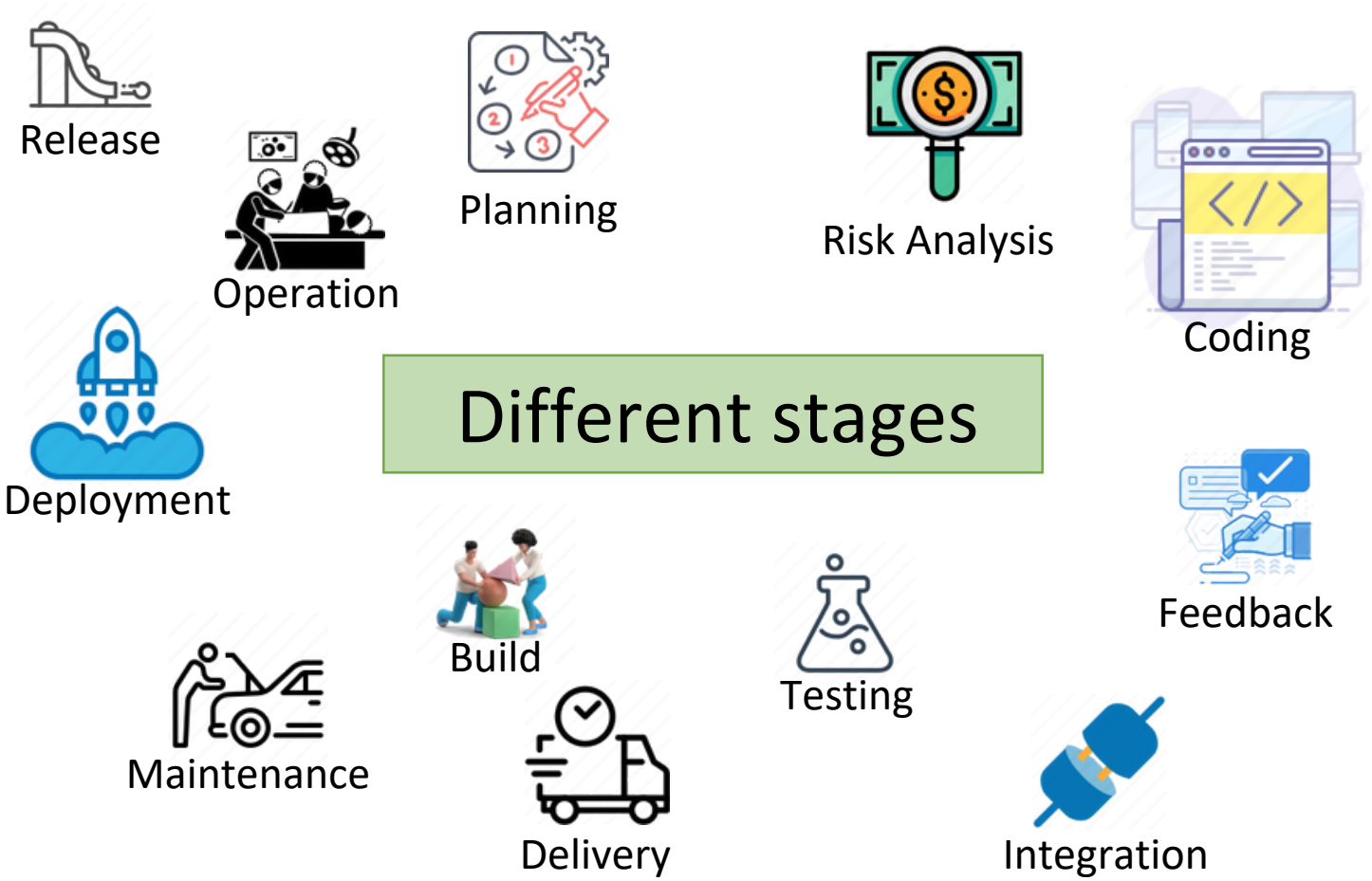
Software Development Life Cycle (SDLC)

- A life cycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].

  - E.g. Banking application
    - Front-end
    - Back-end
    - Databases
    - Analytics
    - endpoints

Software Development Life Cycle (SDLC)

- A life cycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].



Icon src : https://www.iconfinder.com

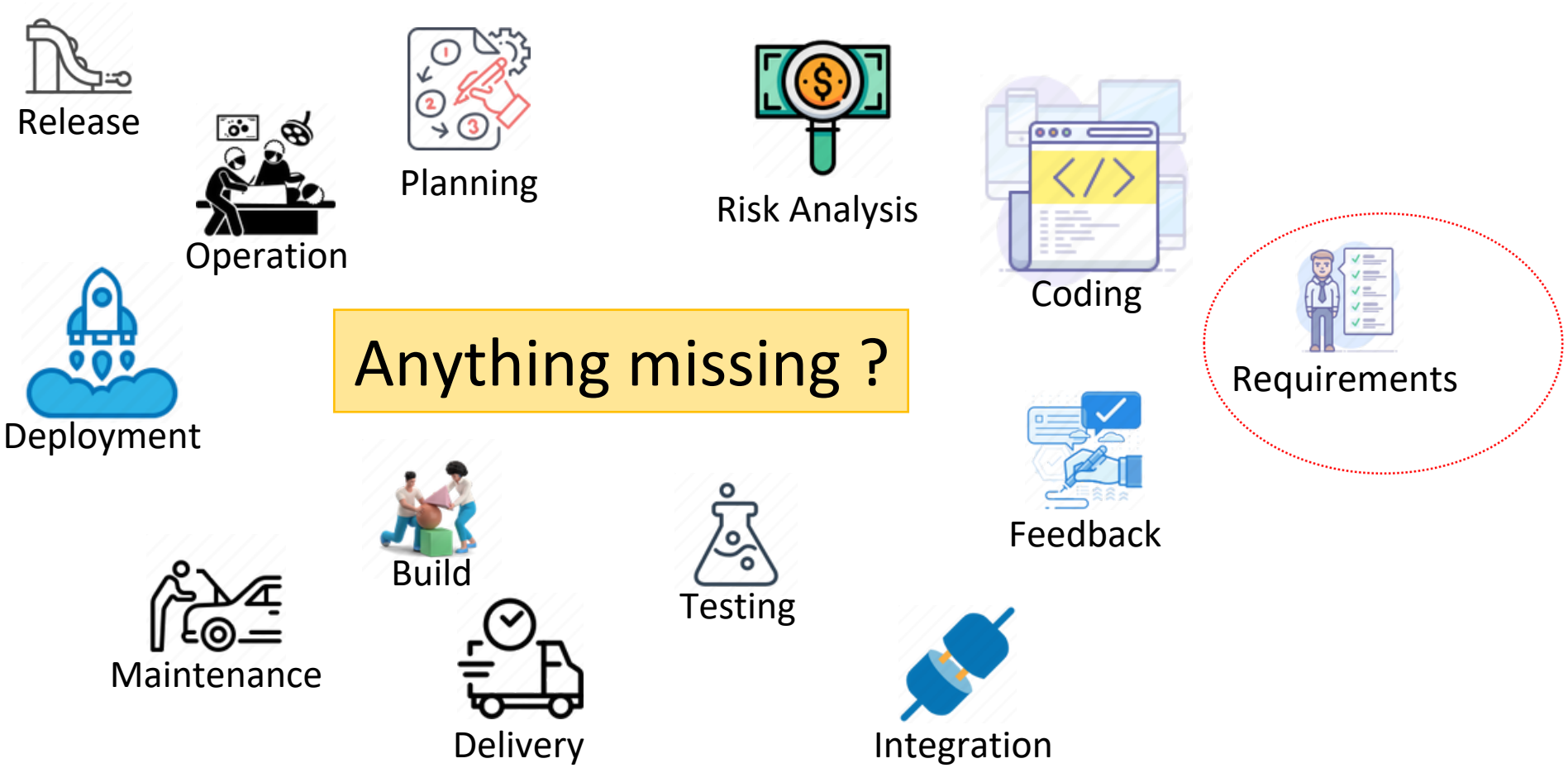
Software Development Life Cycle (SDLC)

- A life cycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].

Icon src : https://www.iconfinder.com

Software Development Life Cycle (SDLC)

- A life cycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].



Icon src : https://www.iconfinder.com

Software Development Life Cycle (SDLC)

- A life cycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].


Release


Operation


Planning


Risk Analysis


Coding


Requirements


Deployment

Can you arrange them?


Feedback


Build


Testing


Maintenance

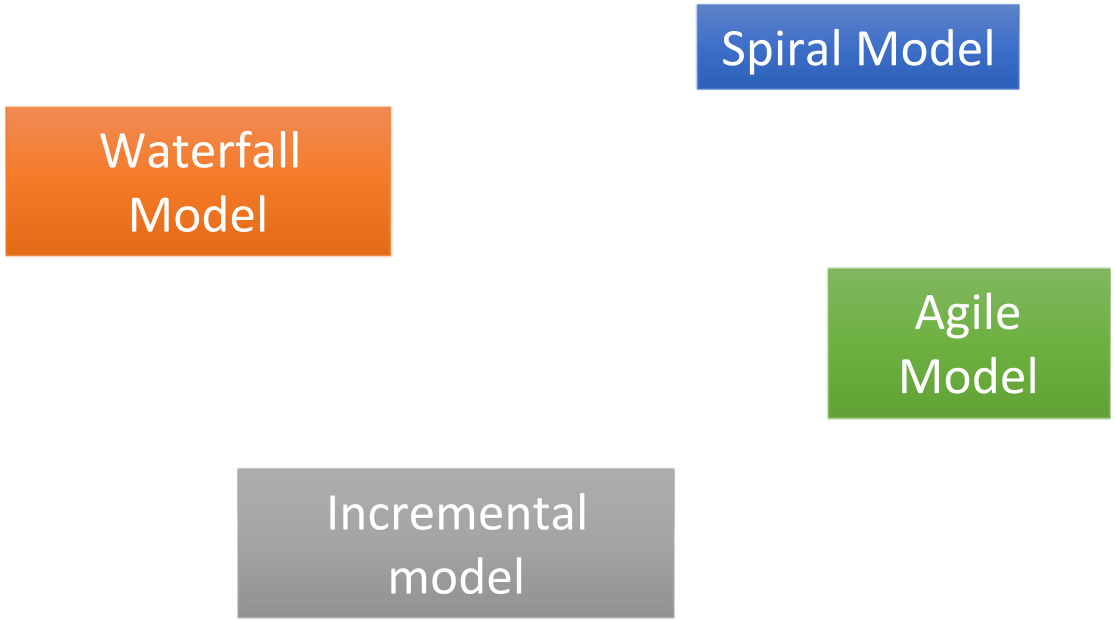

Delivery


Integration

Icon src : https://www.iconfinder.com

Software Development Life Cycle (SDLC)

SDLC models



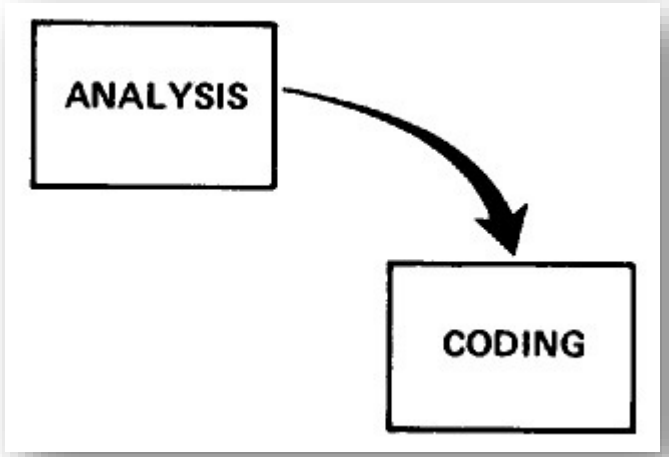
Src: Alshamrani,A., &Bahattab, A. (2015). A comparison between three SDLCmodelswaterfallmodel,spiralmodel,andIncremental/Iterative model. *International Journal of Computer Science Issues (IJCSI),12*(1), 106.

Software Development Life Cycle (SDLC)

Waterfall Models

- Defined as early as ??<year>??

Can you guess ?

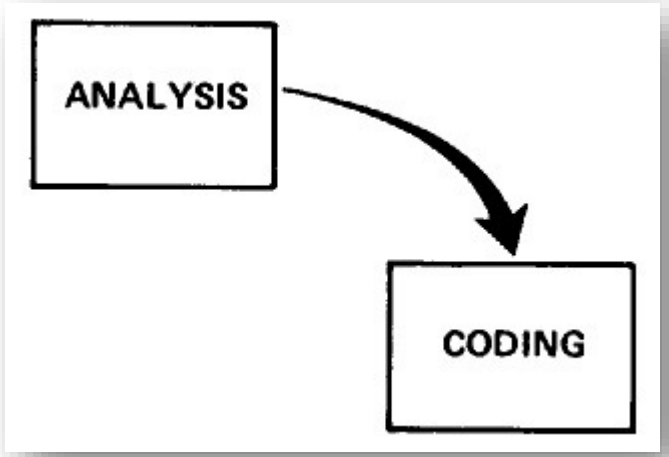


Implementation steps to deliver a small computer program for internal operations

Software Development Life Cycle (SDLC)

Waterfall Models

- Defined as early as 1970 by Dr. Winston W. Royce [royce1970, Davis1988]



Implementation steps to deliver a small computer program for internal operations [royce1970]

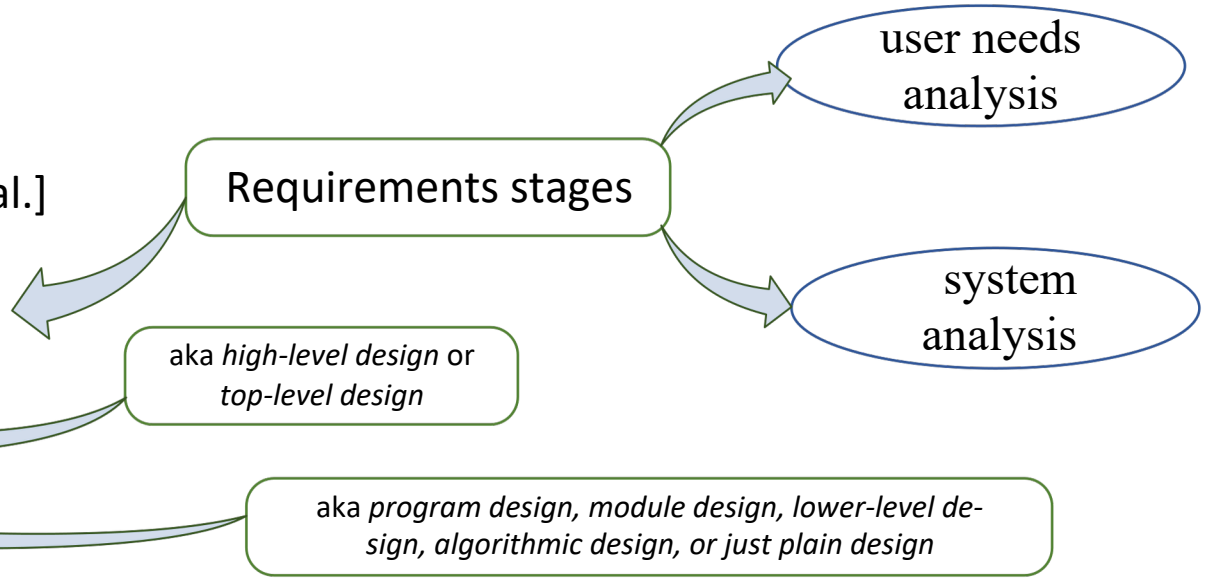
- Later improved in 1976 by Barry W. Boehm [Davis1988,boehm1976]

Src; https://blog.jbrains.ca/assets/articles/royce1970.pdf

Software Development Life Cycle (SDLC)

Waterfall Models

- Most basic stages [Davis et al.]
  - System requirements
  - Software requirements
  - Preliminary Design
  - Detailed Design
  - Code and Debug
  - Test and pre-operation
  - Operations and maintenance



Src; Davis, AlanM.,Edward H. Bersoff, and Edward R. Comer. "A strategyforcomparingalternativesoftwaredevelopmentlife cycle models."IEEETransactions on software Engineering14.10 (1988): 1453-1461.

Software Development Life Cycle (SDLC)

Waterfall Models – General stages

Client

Requirements

Gather the requirements and defining them (users, login, dashboard, purchase cart, payment options)

Programmer

Language, DB, dev platform, other technical requirements…

Planning/De signing

deals with algorithm design, software architecture design, database conceptual schema, logical diagram design, data structure definition, etc.

Writing the code, detecting and removing of existing and potential errors

Coding & Debugging

report, monitor, resolve and retest software components until they reach the quality standards…

Testing

Deliver and deploy the software, ready for production. (e.g. deliver the car…)

Operation

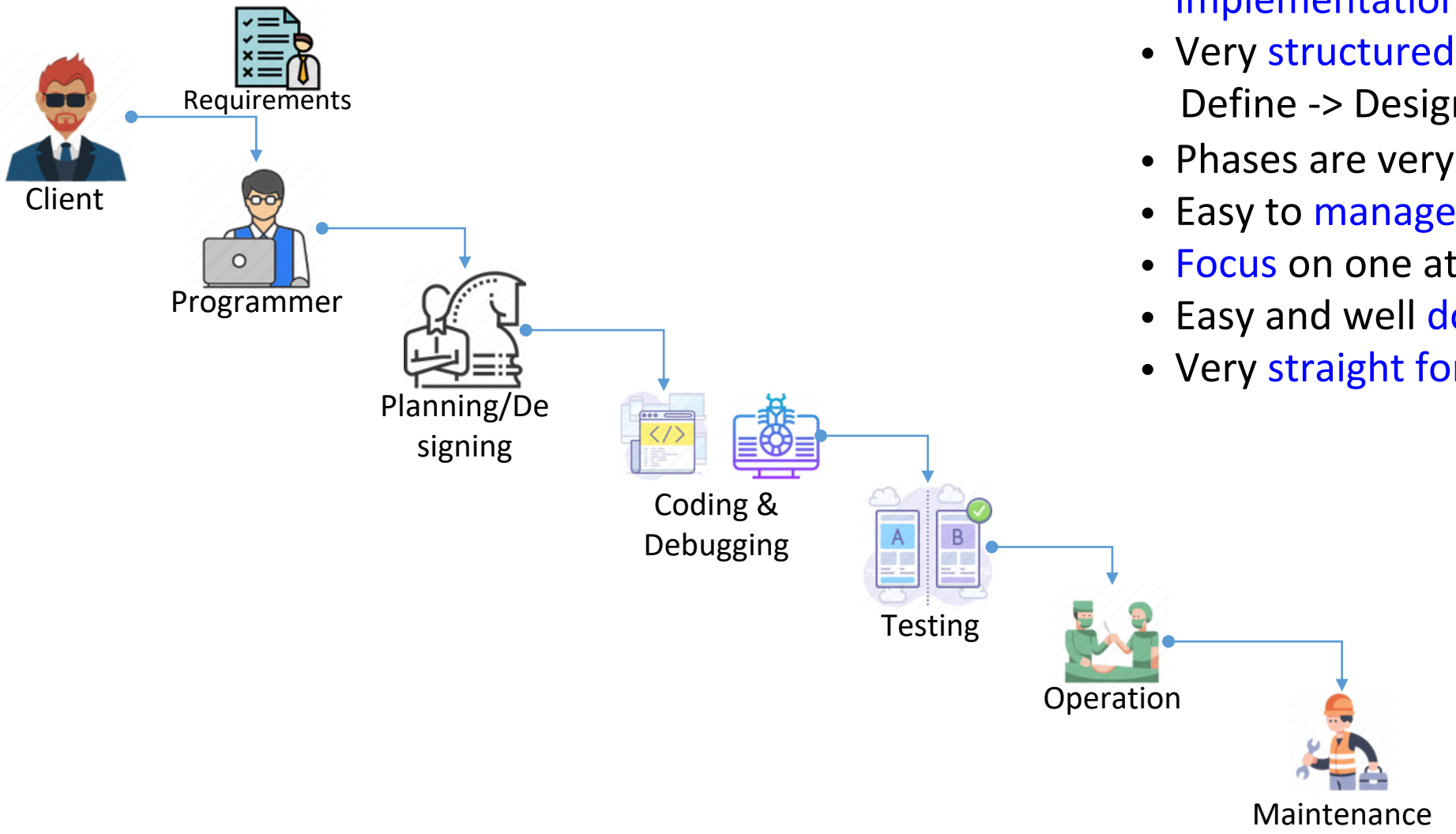
Make sure that the customized car is running smoothly….

Maintenance

Src: Alshamrani,Adel, andAbdullah Bahattab. "A comparison between three SDLCmodelswaterfallmodel,spialmodel,andIncremental/Iterative model." *International Journalof Computer Science Issues (IJCSI)*12.1 (2015): 106.

Software Development Life Cycle (SDLC)

Waterfall Models - **Advantages**

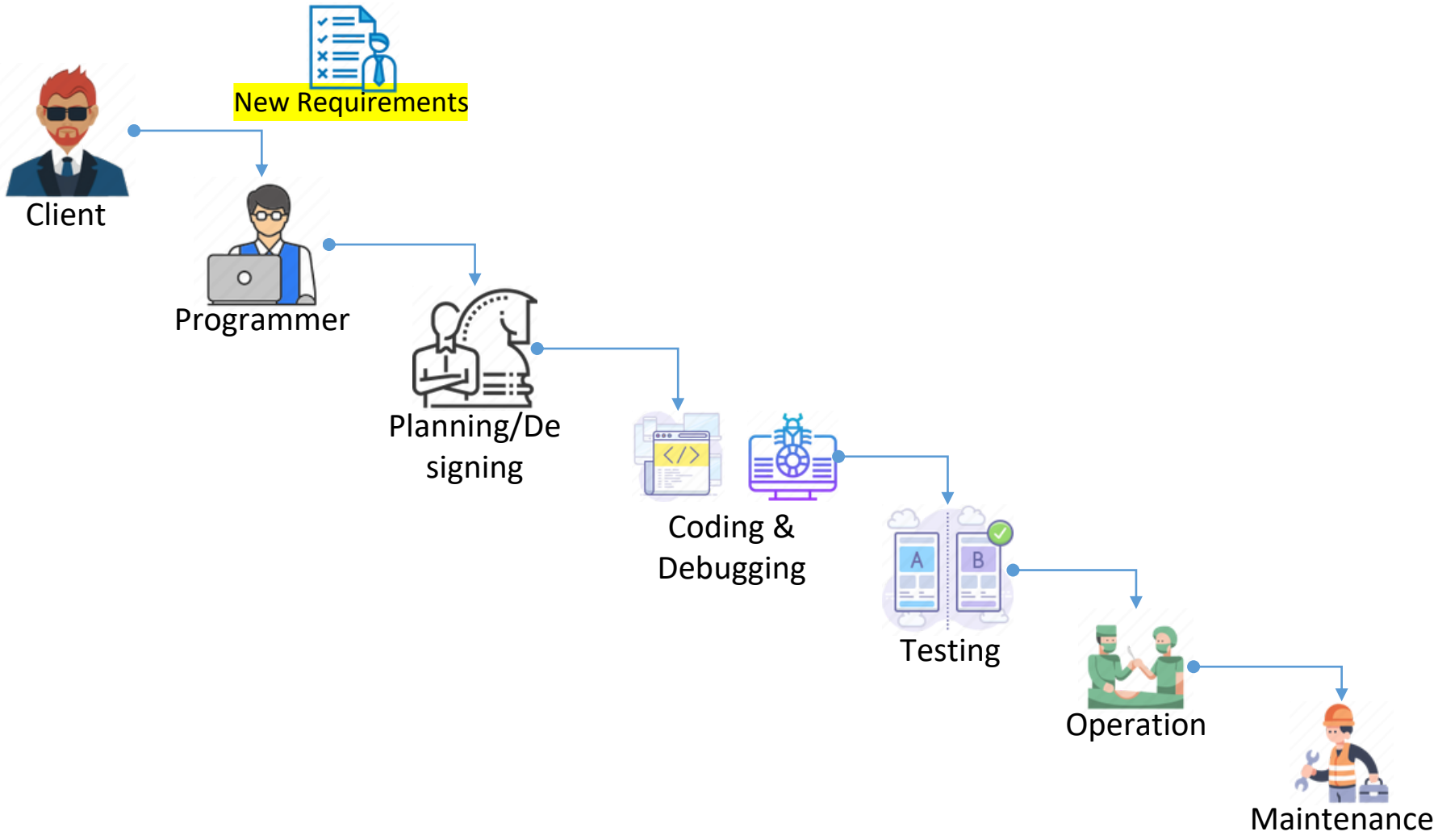


- Simple and easy understanding and implementation
- Very structured organization
  Define -> Design -> Code
- Phases are very clear
- Easy to manage, arrange tasks
- Focus on one at a time
- Easy and well documentation
- Very straight forward steps

Src: Alshamrani,Adel, andAbdullah Bahattab. "A comparison between three SDLCmodelswaterfallmodel,spialmodel,andIncremental/Iterative model." *International Journalof Computer Science Issues (IJCSI)*12.1 (2015): 106.

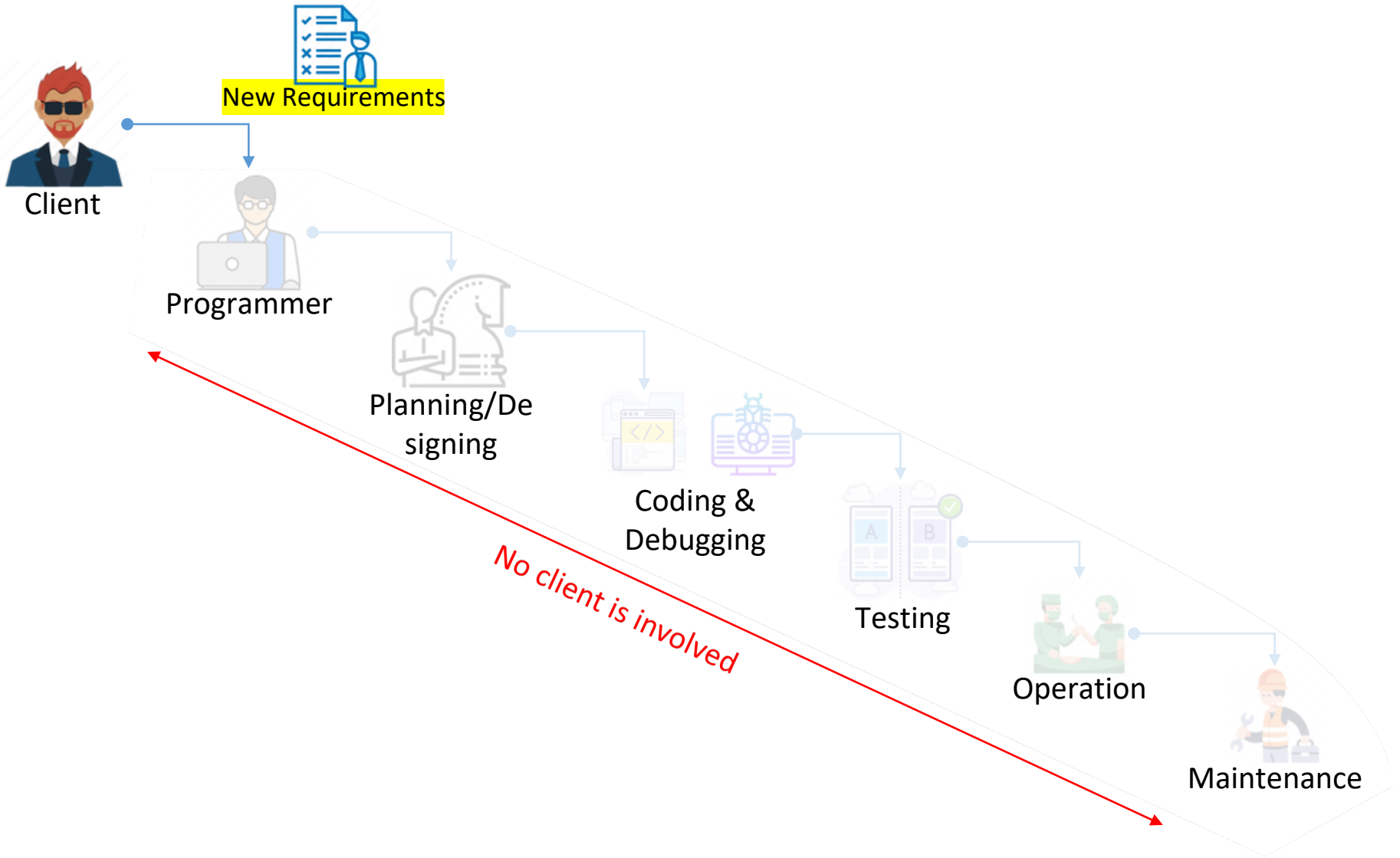
Software Development Life Cycle (SDLC)

Waterfall Models



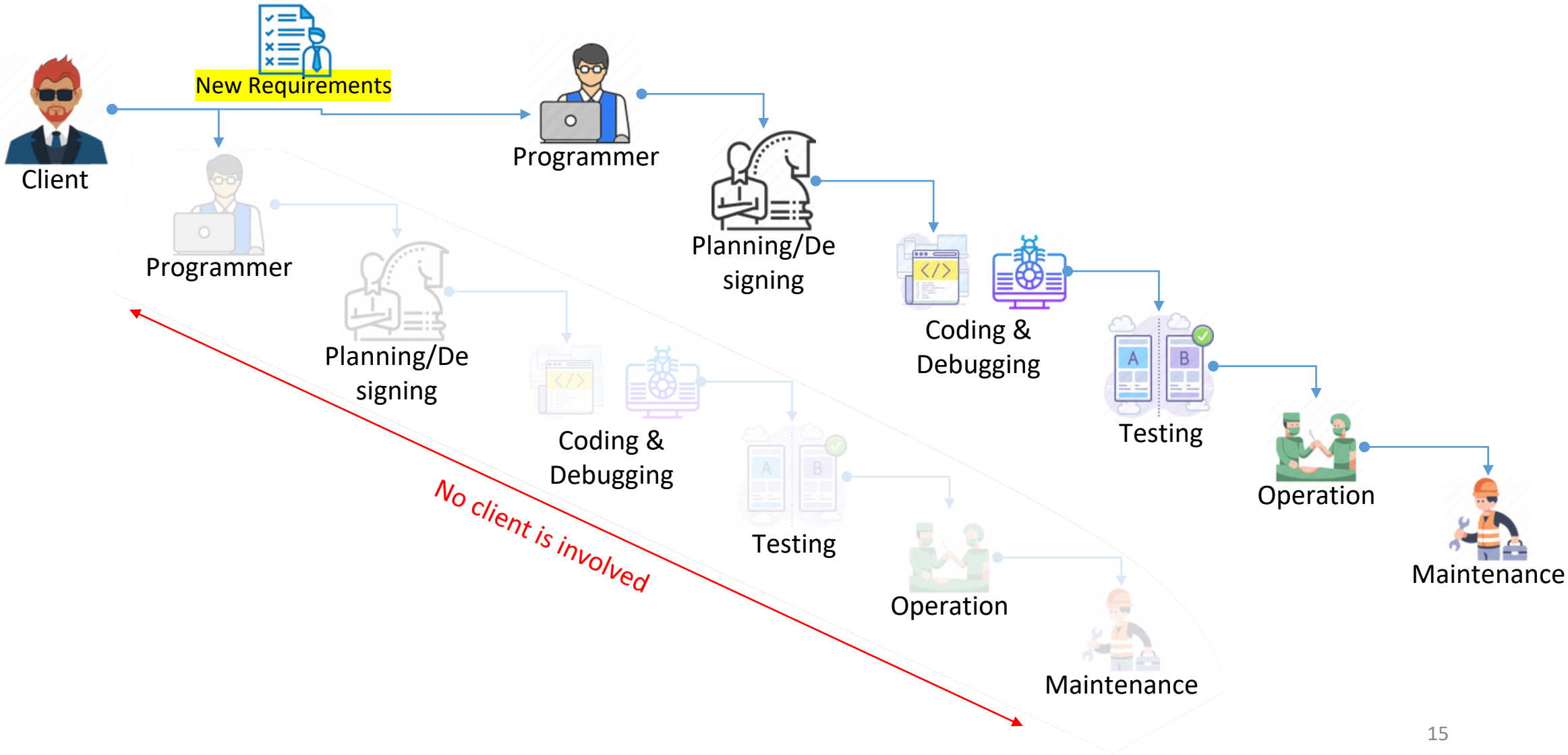
Software Development Life Cycle (SDLC)

Waterfall Models



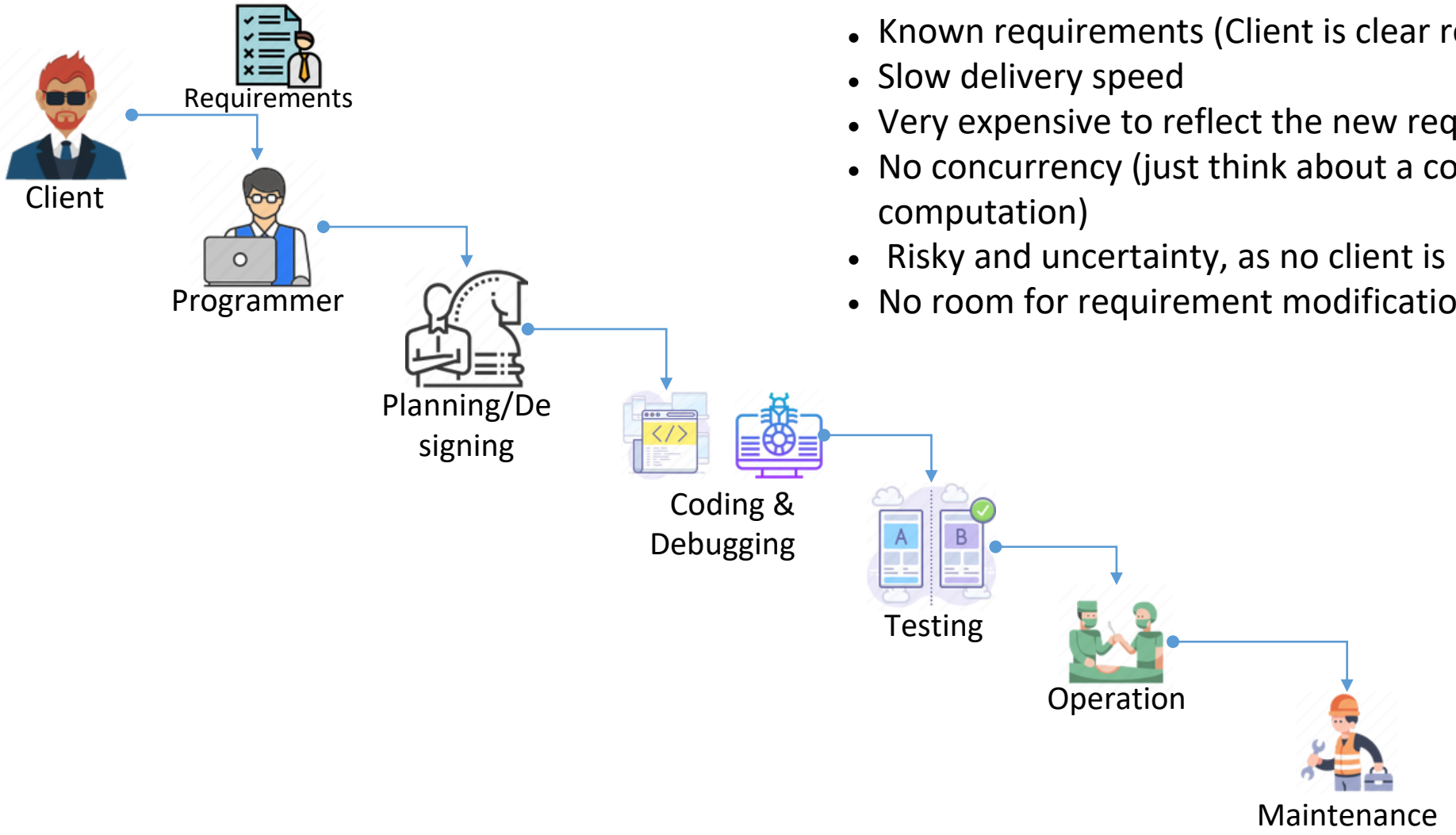
Software Development Life Cycle (SDLC)

Waterfall Models



Software Development Life Cycle (SDLC)

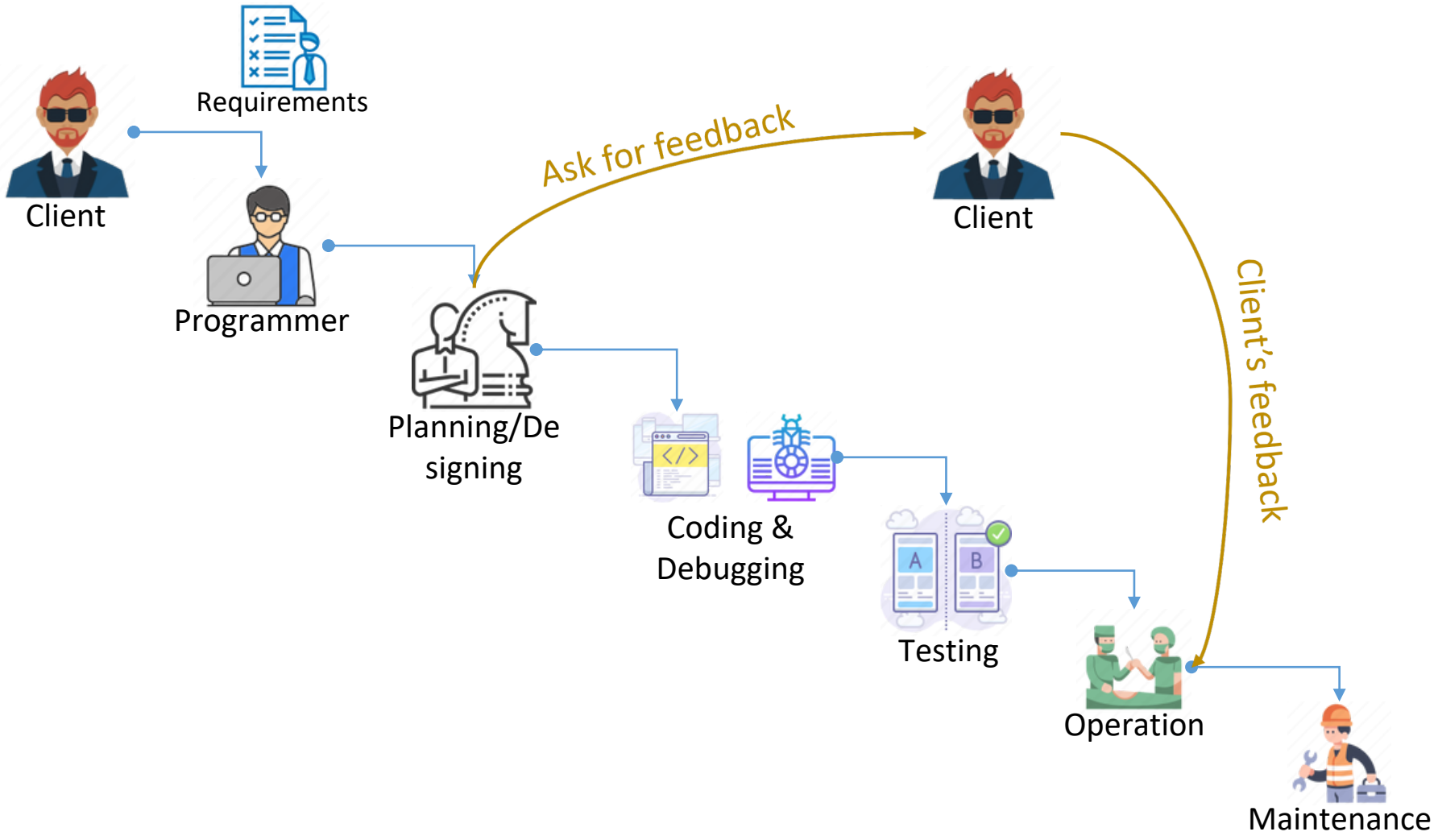
Waterfall Model - **Disadvantages**



- Known requirements (Client is clear regarding their requirements)
- Slow delivery speed
- Very expensive to reflect the new requirements
- No concurrency (just think about a computer with single- thread computation)
- Risky and uncertainty, as no client is involved in other stages
- No room for requirement modification in later stages

Software Development Life Cycle (SDLC)

Agile Model



Software Development Life Cycle (SDLC)

Agile Model

Src: https://dictionary.cambridge.org/dictionary/english/agile

Software Development Life Cycle (SDLC)

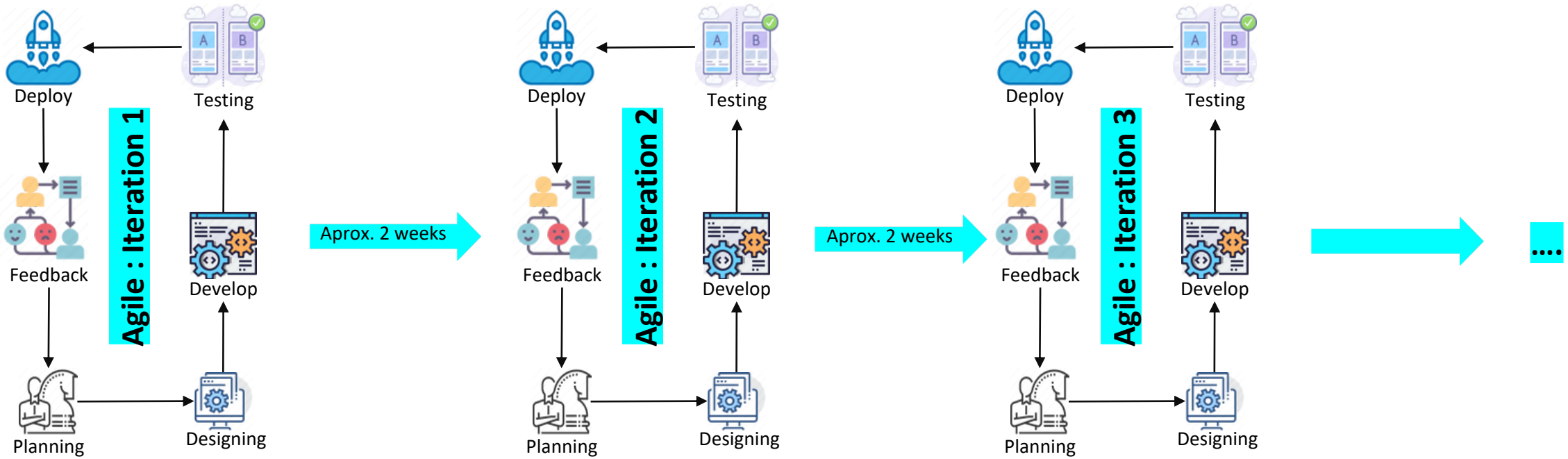
Agile Model

- Address the challenge of an unpredictable world
- Focus on individuals and their creativity
  - rather than on processes and tools
- Promote quick response to
  - changing environments
  - changes in user requirements

Its all about “about feedback and change” [Williams2003]

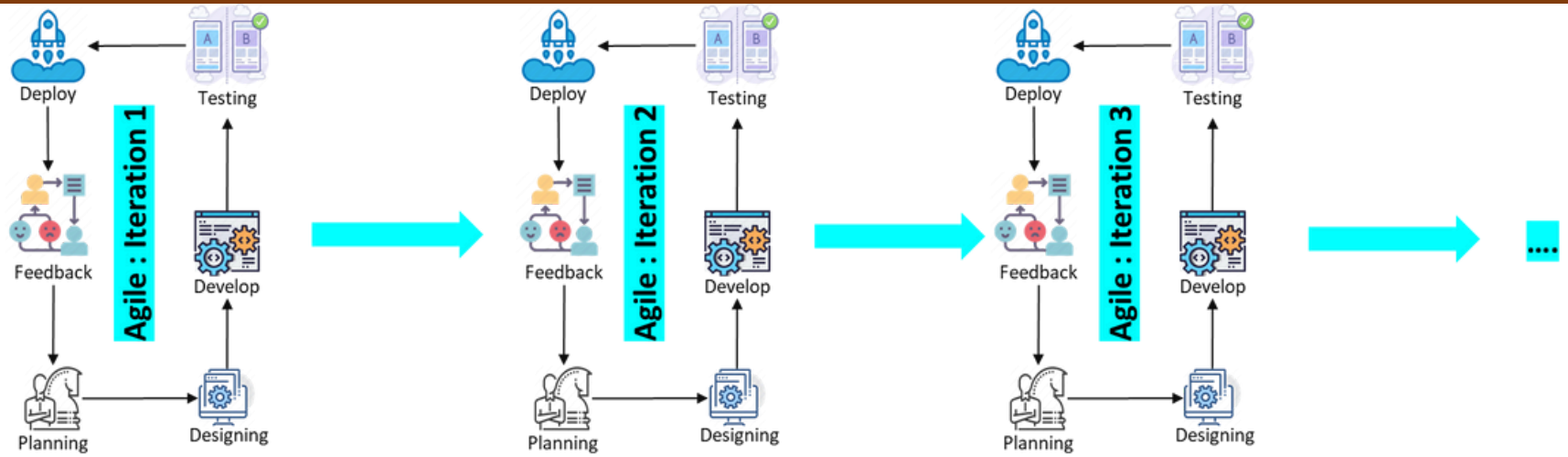
Software Development Life Cycle (SDLC)

Agile Model



Software Development Life Cycle (SDLC)

Agile Model



What makes a development method an agile one? [6]

- Breakdown the whole application to different sub-services
- **Incremental** : small software releases, with rapid cycles
- **cooperative** : customer and developers working constantly together with close communication
- **Straight forward**: the method it self is easy to learn and to modify
- **adaptive :** able to make last moment changes
- No monolithic application development

Software Development Life Cycle (SDLC)

Agile Model – Principles

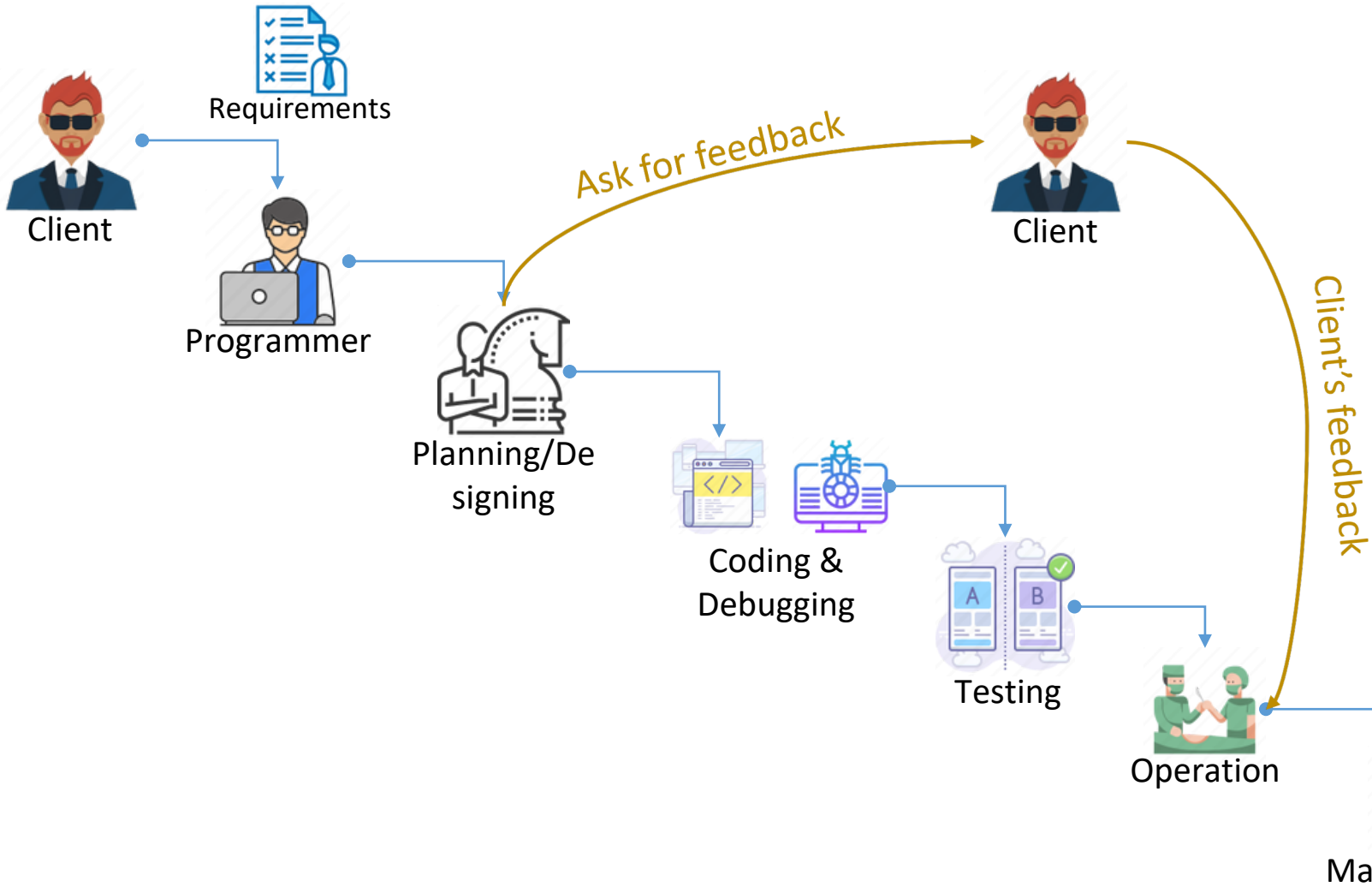
Principles of Agile development model [5]

1. Client's satisfaction becomes higher priority
2. Welcome changing requirements
3. Frequent interaction with clients
4. Frequent working software delivery
5. Provide motivated environment and support to individuals
6. Face-to-face conversation(now probably on virtual mode)
7. Working software becomes the primary measure of progress
8. Sustainable development (realistic expectations, pace, consistency, prioritization, etc.)
9. Continuous attention to technical excellence and good design enhances agility
10. Simplicity–Maximize the amount of work not done
11. The best architectures, requirements, and designs emerge from self-organizing teams

Src: http://agilemanifesto.org/principles.html

Software Development Life Cycle (SDLC)

Agile Model - **Advantages**

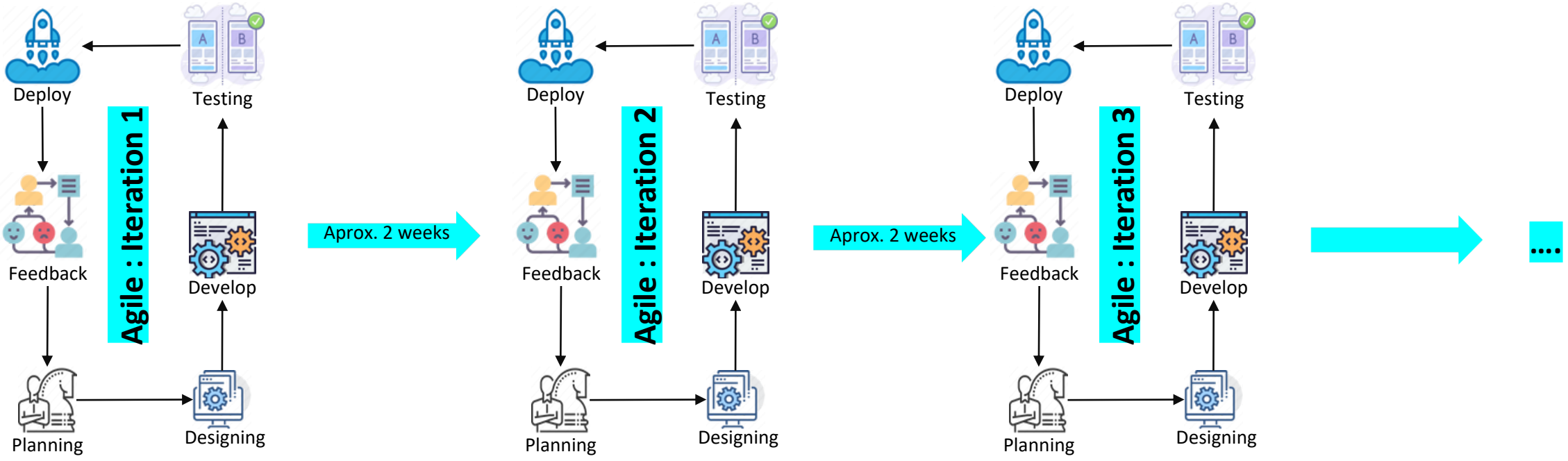


1. Persistent software delivery
2. Quicker release through fixed, reliable and short-term schedules
3. Quick adoption of new features
4. Increased client's satisfaction
5. Frequent interaction
6. Continuous improvement
7. Increased transparency through tighter collaboration
8. Improve product quality and predictability
9. Reduced risks

Software Development Life Cycle (SDLC)

Seems this is a perfect model for software development.

then why we need DevOps ?



Software Development Life Cycle (SDLC)

Agile Model:

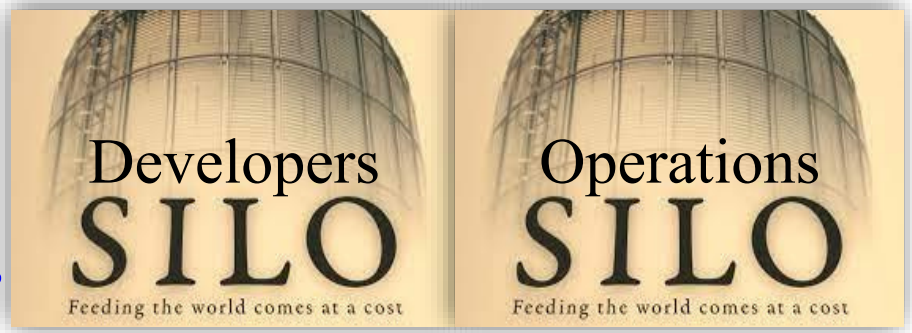
Disadvantages:

1. Developer computer is used mainly for testing of those features

2. Test and production environment configuration mismatch

3. Usually not tested in production environment

4. Developers team and operations team are in silos



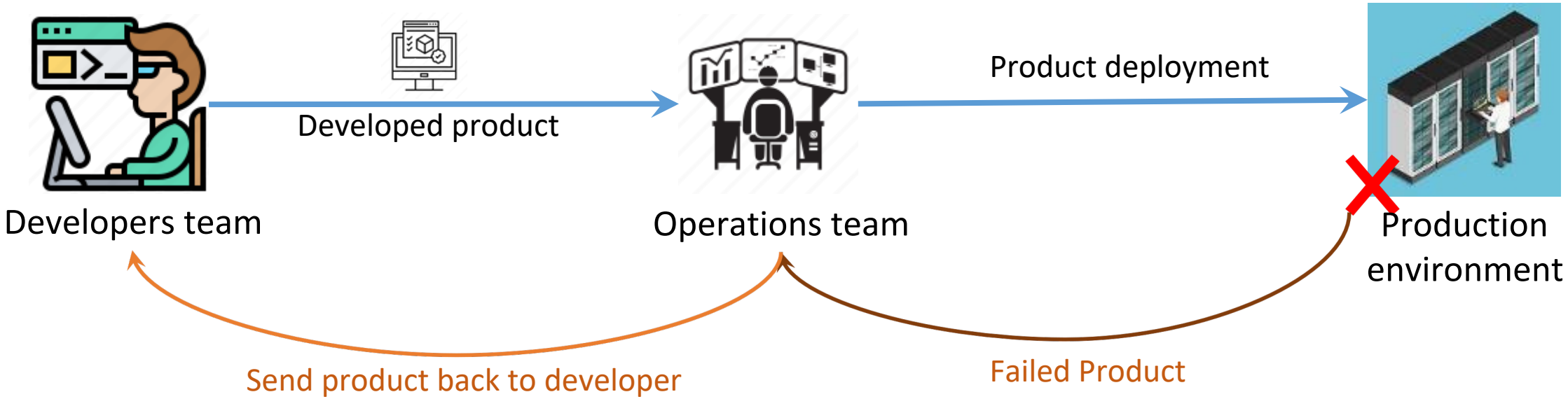
5. Continuous involvementof all the stakeholders.

Img src: https://www.dglobe.com/business/agriculture/4972636-Free-showing-of-Silo-movie-is-March-14

Software Development Life Cycle (SDLC)

Agile Model:

Disadvantages:



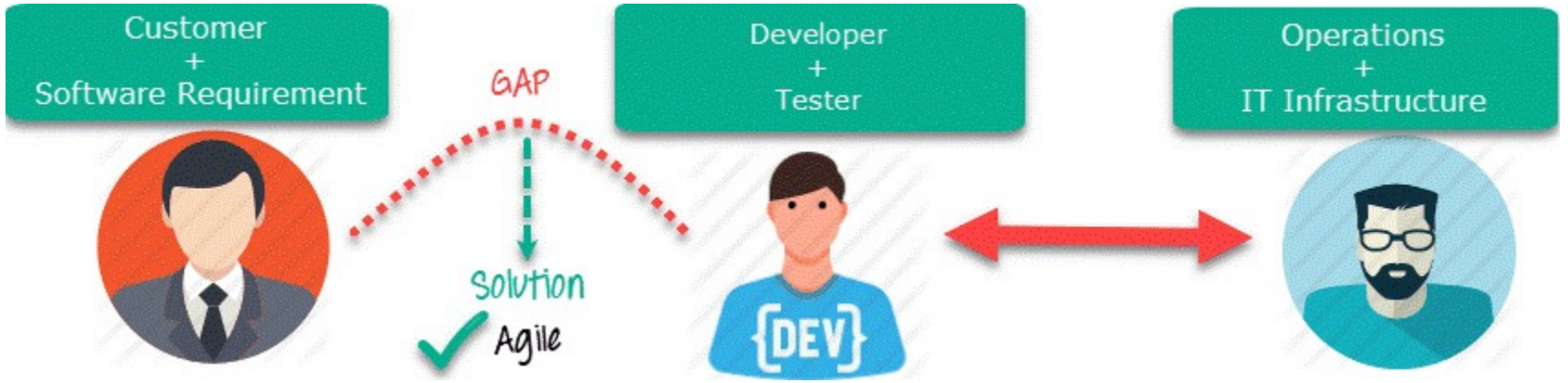
Img src: https://www.dglobe.com/business/agriculture/4972636-Free-showing-of-Silo-movie-is-March-14

Software Development Life Cycle (SDLC)

Agile Model:

Disadvantages:

Agile addresses gaps in **Customer** and **Developer** communications



DevOps addresses gaps in **Developer**and **IT Operations** communications

Img src: https://www.guru99.com/agile-vs-devops.html

Any question so far…..?

DevOps

Presentation Topics

Topic 1: The Silo Effect: When Dev and Ops Don't Talk

- What are "silos" in IT organizations?
- Focus: Analyze the cultural and organizational divide between Development and Operations teams.
- Conflicting goals
- How this leads to slow releases
- The human cost: blame, stress, and burnout.

Topic 2: The "It Works on My Machine!" Problem

- Why do developers mainly test on their own computers?
- What causes configuration mismatches between test, staging, and production?
- Real-world consequences
- Historical "solution": Long, manual release checklists and blame games.