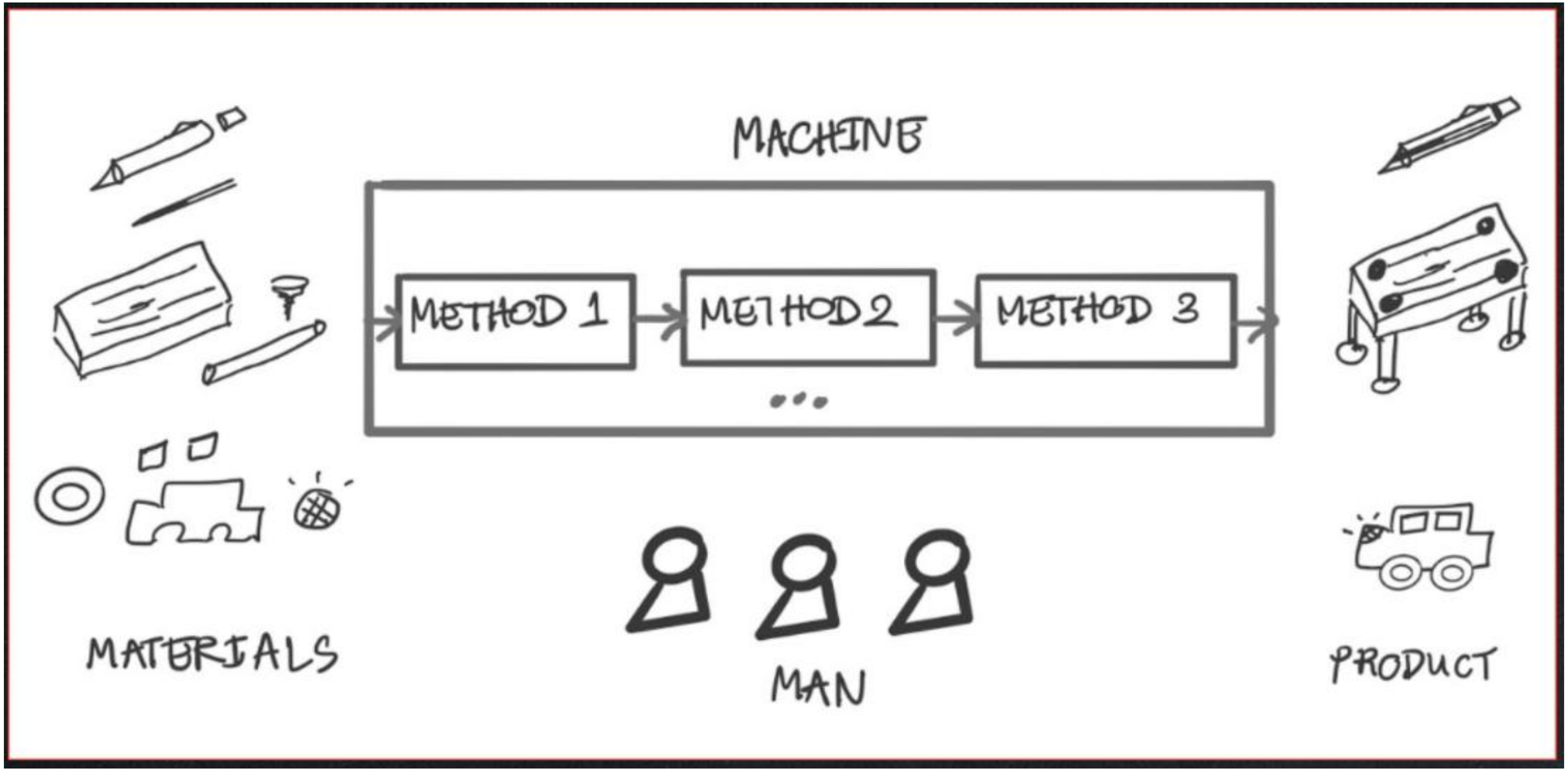
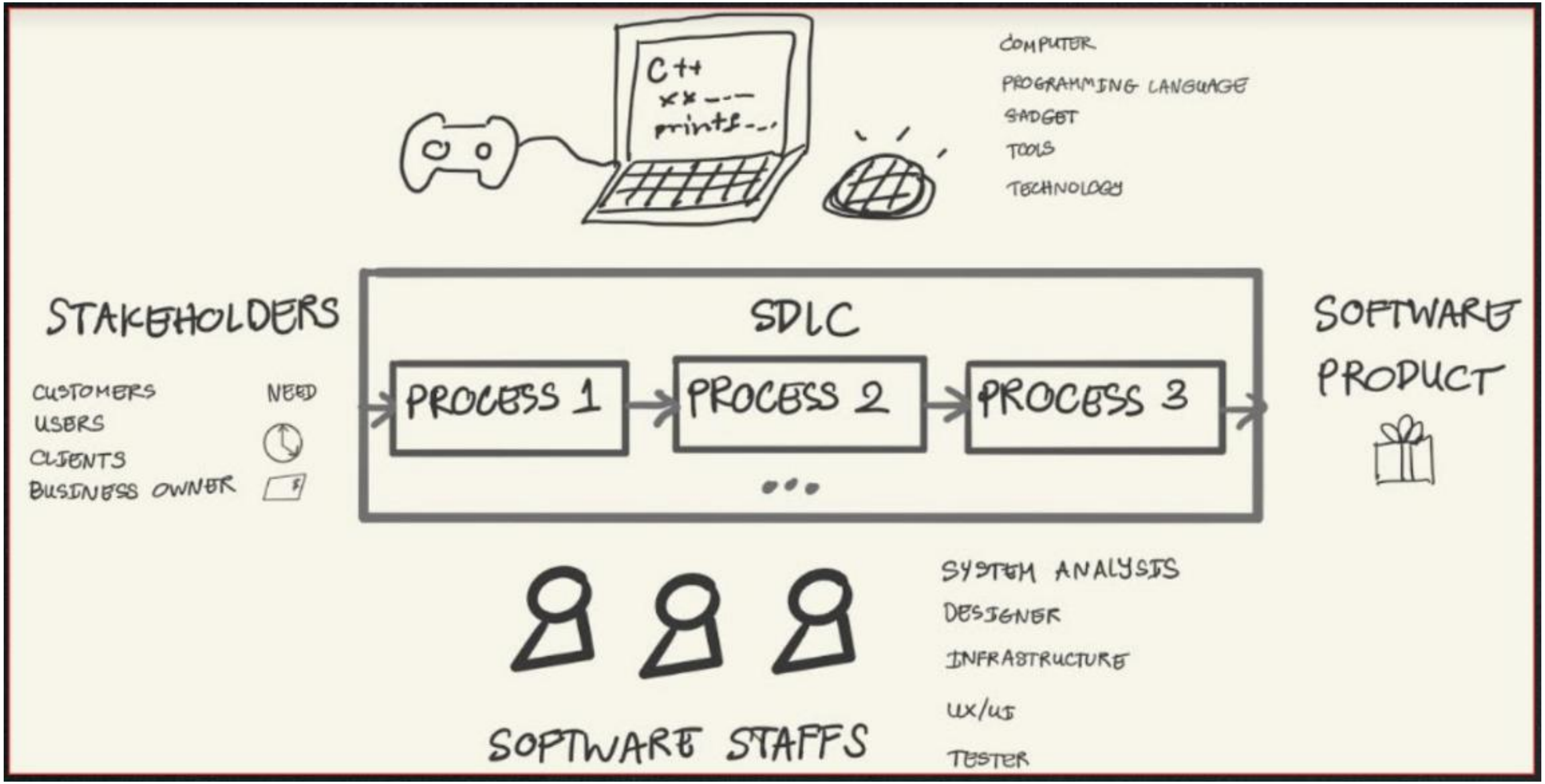
Chapter 1.1

Understanding Requirement

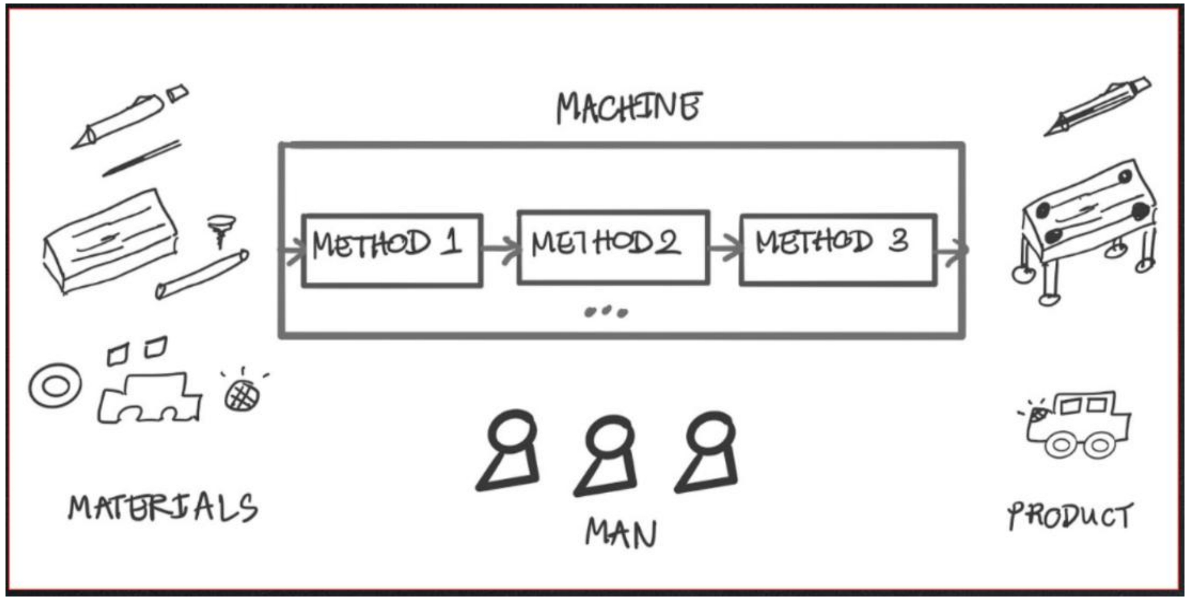


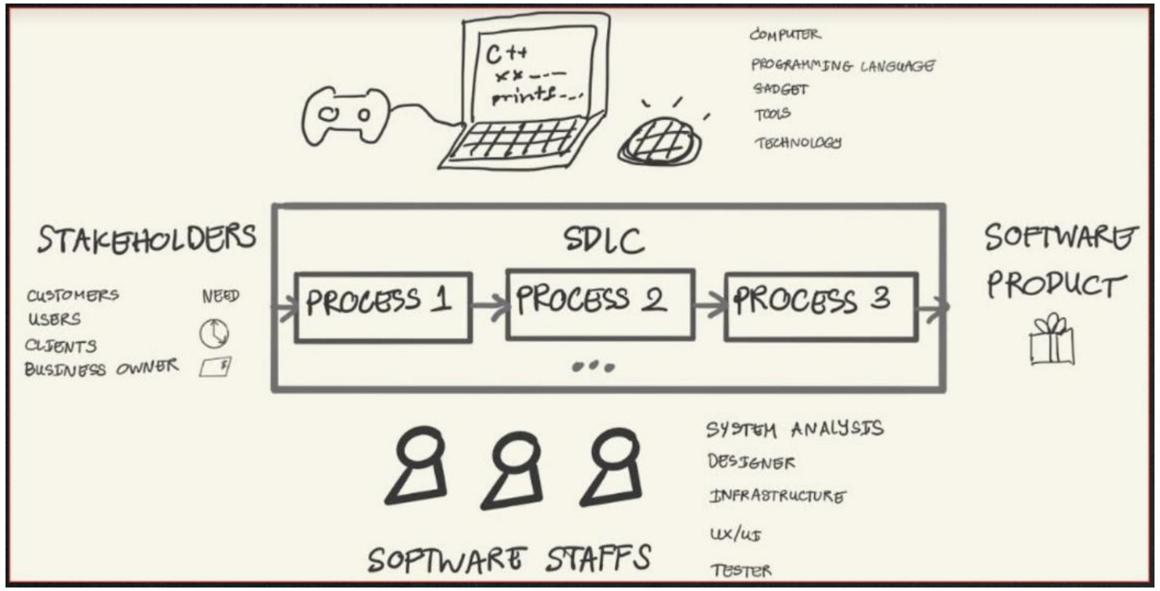
In a production, what is required to build a product?



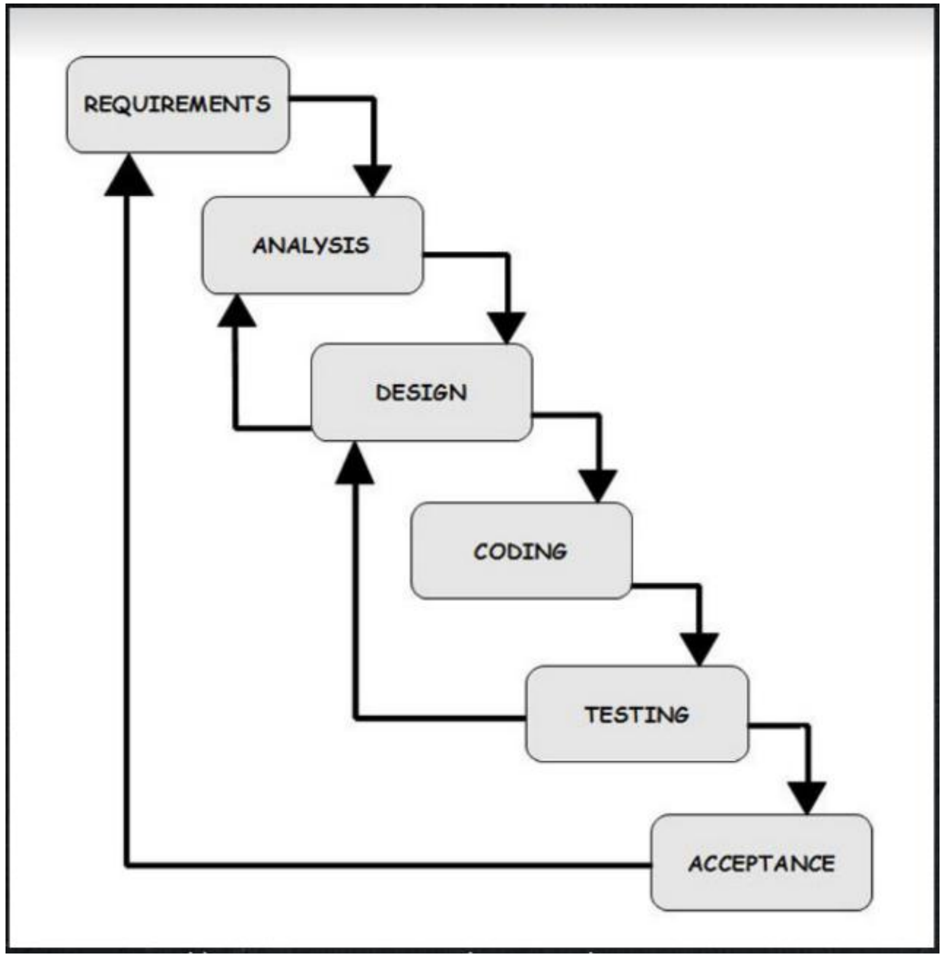
In a software development,
what is required to build a software product?

Similarities? | Differences?

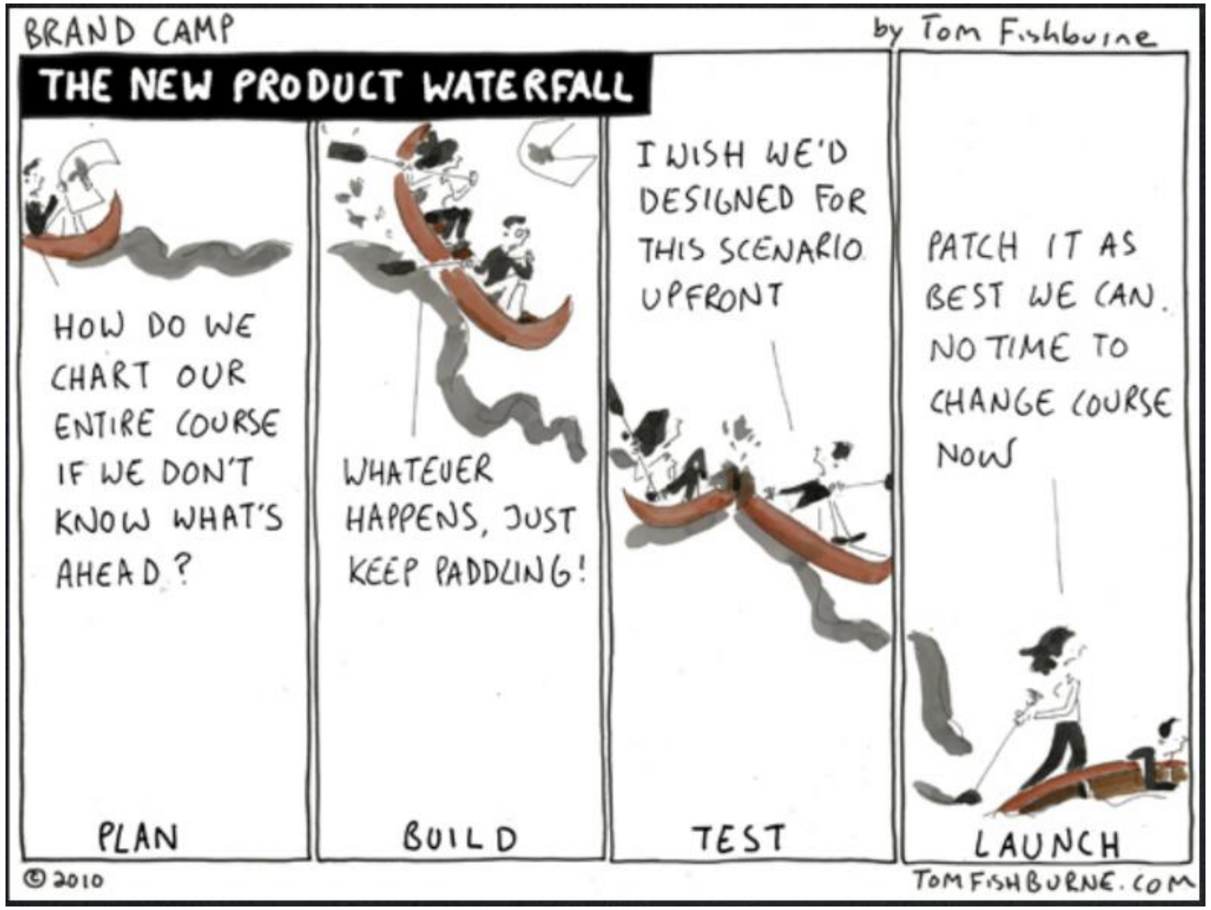




Waterfall model

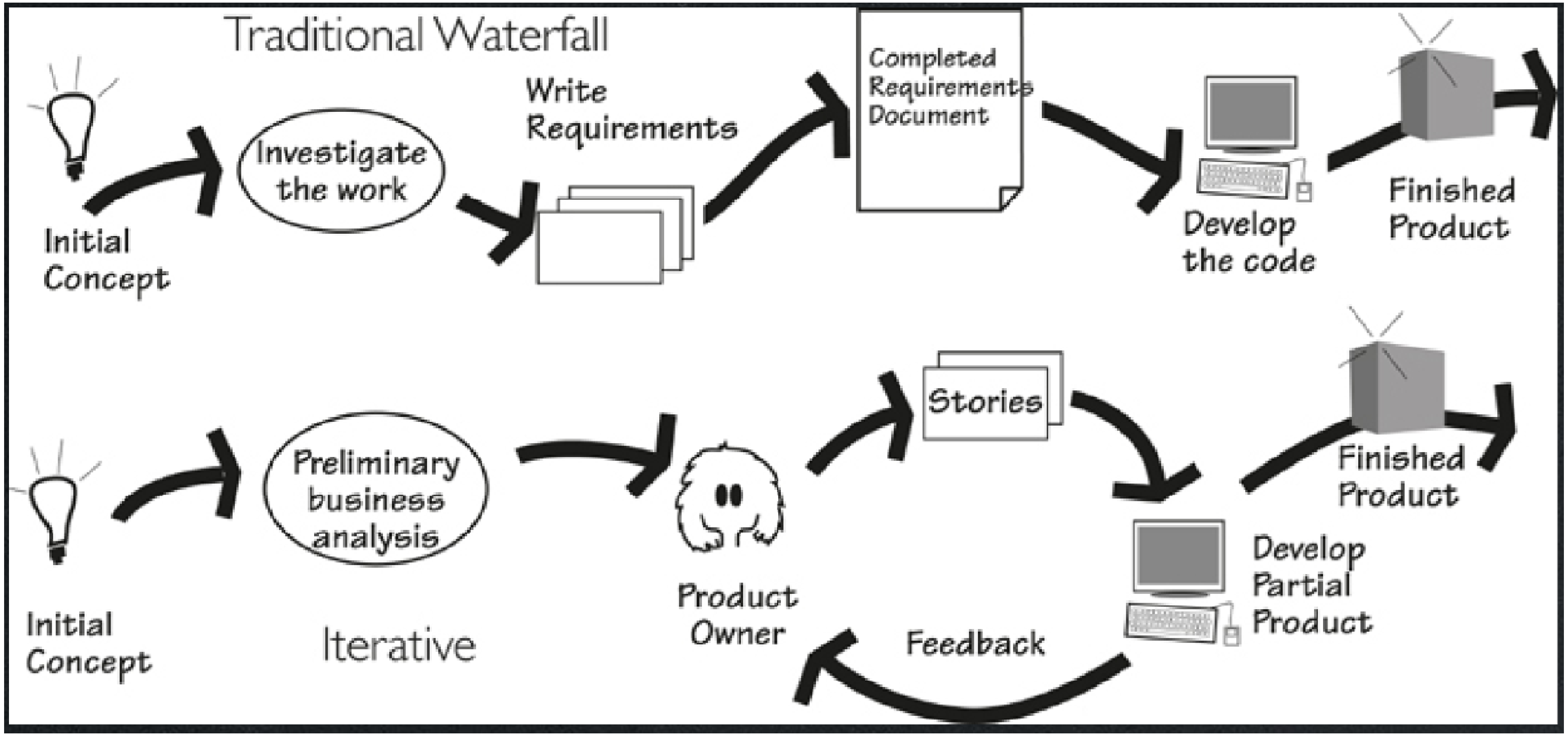


http://www.buzzle.com/articles/comparison-between-waterfall-model-and-spiral-model.html

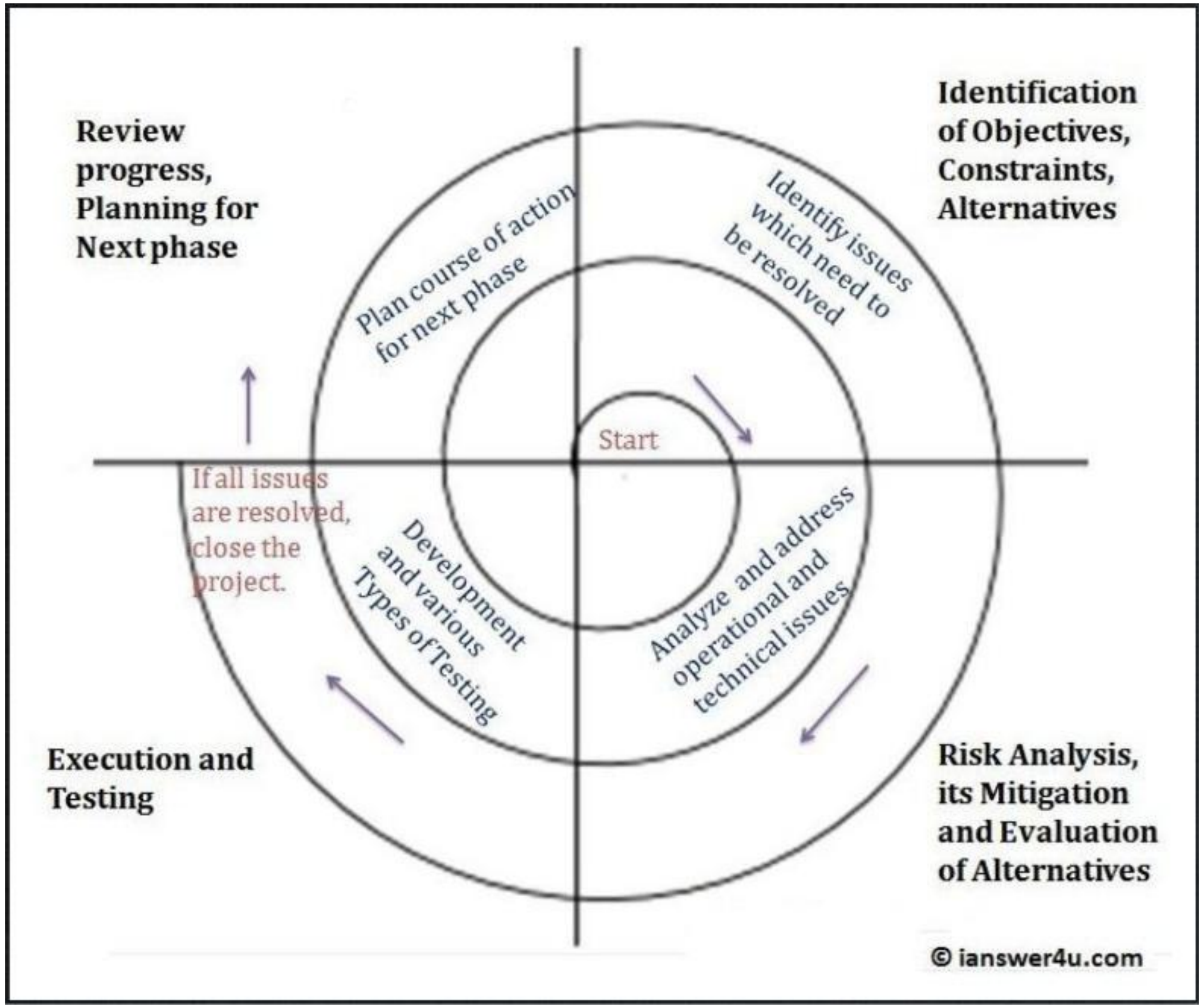

BRAND CAMP
by Tom Fishburne
THE NEW PRODUCT WATERFALL
HOW DO WE CHART OUR ENTIRE COURSE IF WE DON'T KNOW WHAT'S AHEAD?
WHATEVER HAPPENS, JUST KEEP PADDLING!
I WISH WE'D DESIGNED FOR THIS SCENARIO UPFRONT
PATCH IT AS BEST WE CAN. NO TIME TO CHANGE COURSE NOW
PLAN
BUILD
TEST
LAUNCH
© 2010
TOMFISHBURNE.COM

http://tomfishburne.com/2010/04/the-new-product-waterfall.html

Waterfall vs. Iterative

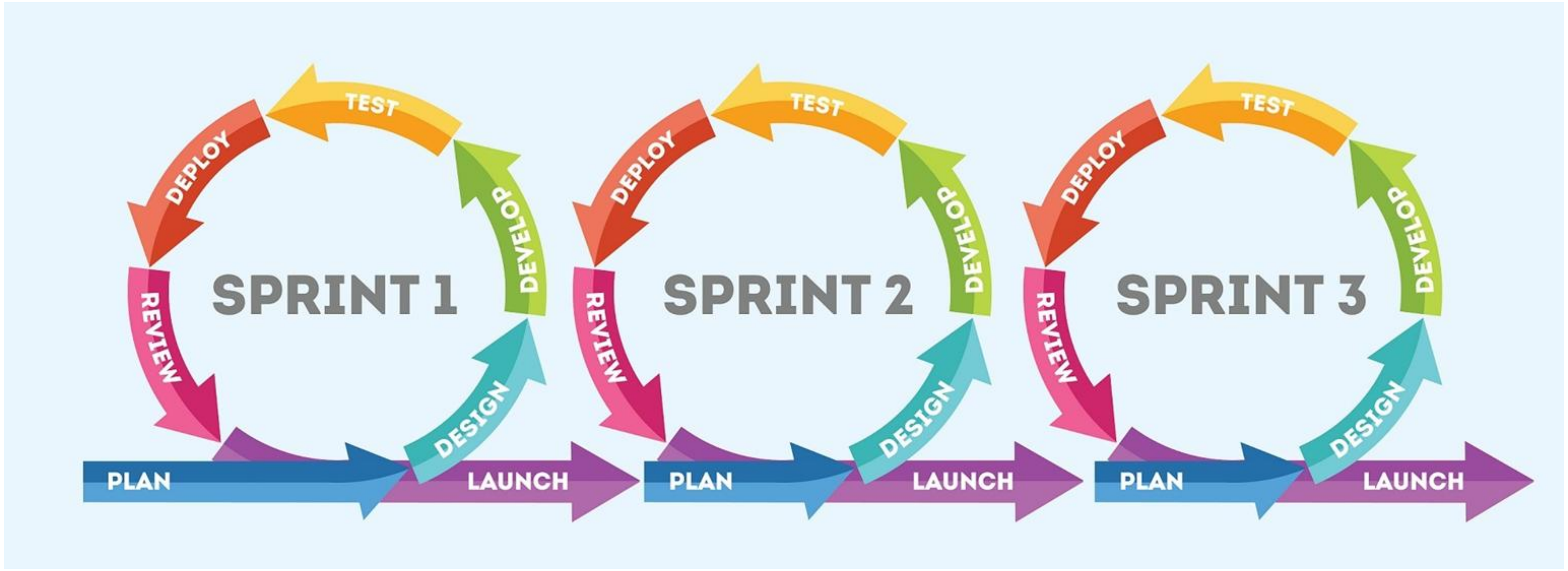


Spiral Model


Review progress, Planning for Next phase
Identification of Objectives, Constraints, Alternatives
Plan course of action for next phase
Identify issues which need to be resolved
Start
If all issues are resolved, close the project.
Development and various Types of Testing
Analyze and address operational and technical issues
Execution and Testing
Risk Analysis, its Mitigation and Evaluation of Alternatives
© ianswer4u.com

http://www.ianswer4u.com/2011/12/spiral-model-advantages-and.html#axzz2UeLxfHYB

Agile

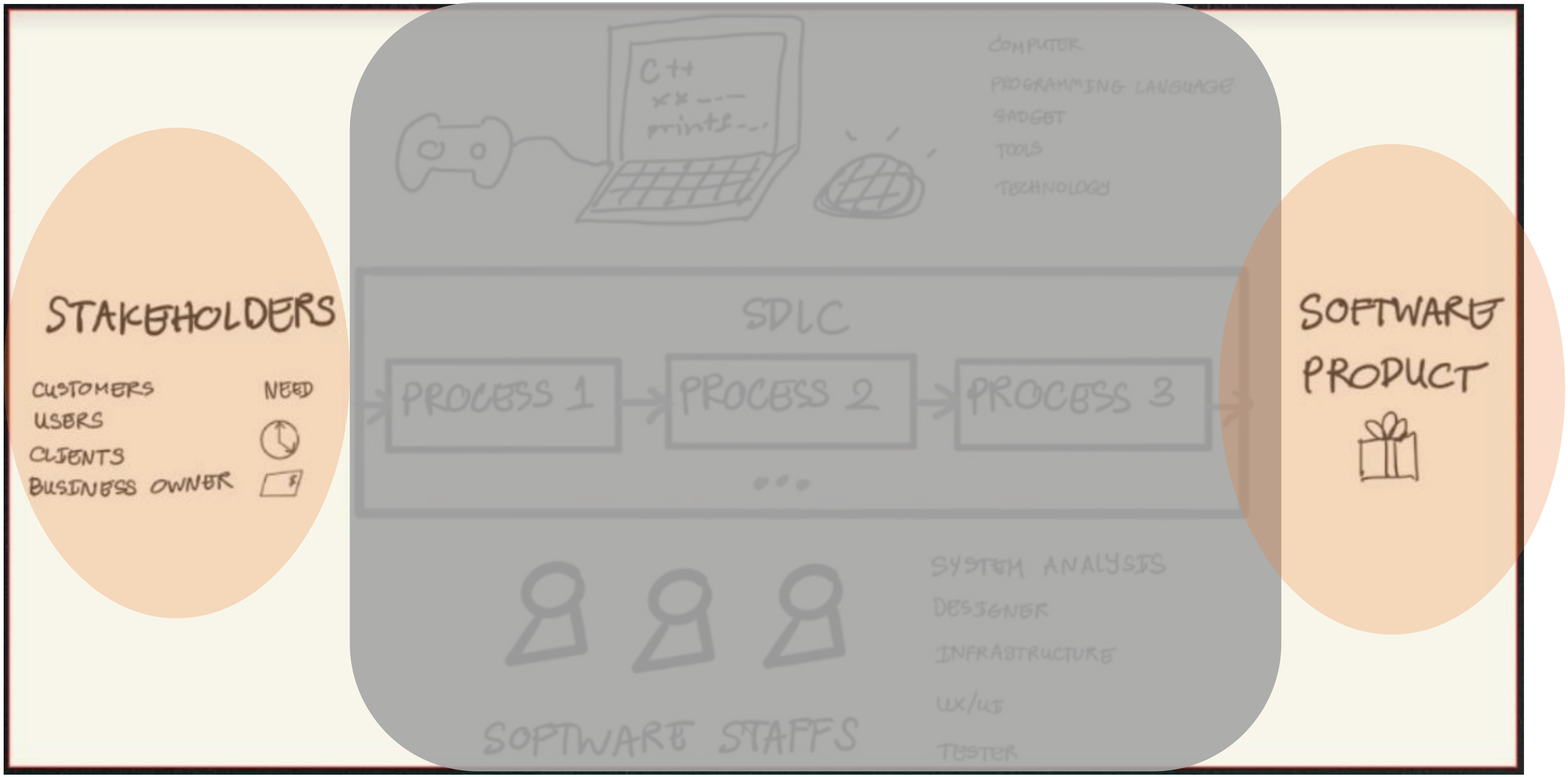


https://www.soldevelo.com/blog/is-agile-always-the-best-solution-for-software-development-projects/

What are Requirements?

A requirement is something
a product **must do** or *a quality* it **must have**

in order to accomplish
the **goals** *of users* or *organizations*.


STAKEHOLDERS
CUSTOMERS
USERS
CLIENTS
BUSINESS OWNER
NEED
C++
printf
COMPUTER
PROGRAMMING LANGUAGE
GADGET
TOOLS
TECHNOLOGY
SDLC
PROCESS 1
PROCESS 2
PROCESS 3
SYSTEM ANALYSIS
DESIGNER
INFRASTRUCTURE
UX/UI
TESTER
SOFTWARE STAFFS
SOFTWARE PRODUCT

What are Requirements?

It is about **needs** and **problems**,

not about solutions!

Why do we care about the requirements?

At the beginning, we **do not** care

- the **programming language** ,
- the development **tools** , or
- the development **process**

We will build the **right product!**

(Right to the mind of the **stakeholders**, especially the real **users**)

Why do we care about the requirements?

"The *hardest* single part of building a software system is

>> deciding what to build.

No part of the work so **cripples** the resulting system if done wrong.

No other part is more difficult to **rectify** later."

Fred Brooks

We create the software *before* there is the software!

Why do we care about the requirements?

What to build?

- **what** the product has to **do?**
- **how** it will be used, by **whom** it will be **used?**
- how it **fits into** the larger picture of the **organization**, etc.?
- which **constraints** it must satisfy?

Why do we care about the requirements?

The **accomplished** software are those

where the **developers understand** about ……….?

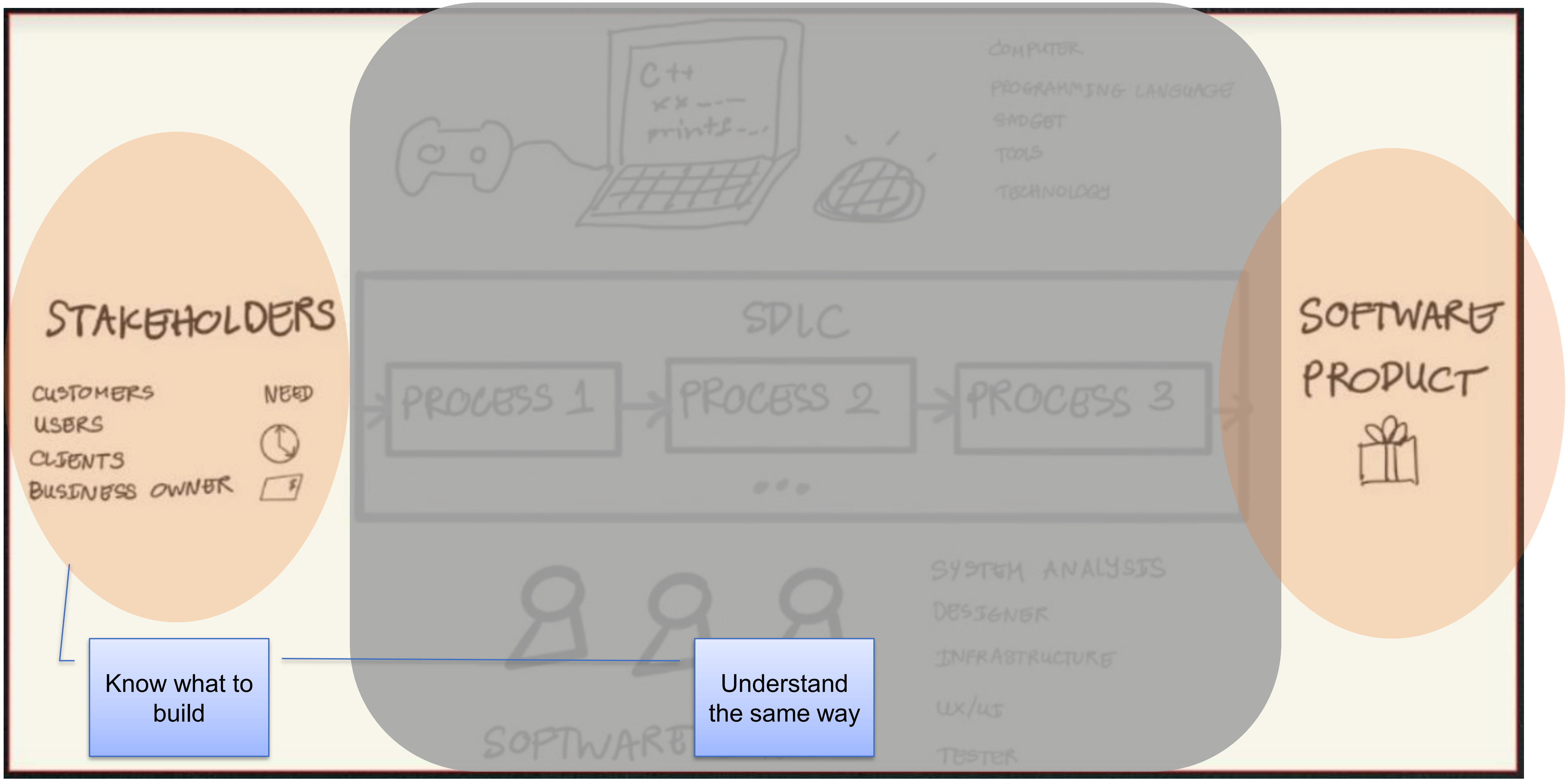
- **what** the product is intended to accomplish for its **users**
- and **how** it must accomplish that purpose.

WHAT DO THE USERS WANT ?

Why do we care about the requirements?

- How do the developer **know** what do the **users want** ?
- How do the developer come to the **correct understanding** of the requirement?
- Do they make sure that the client also understand them

in the **same way**?

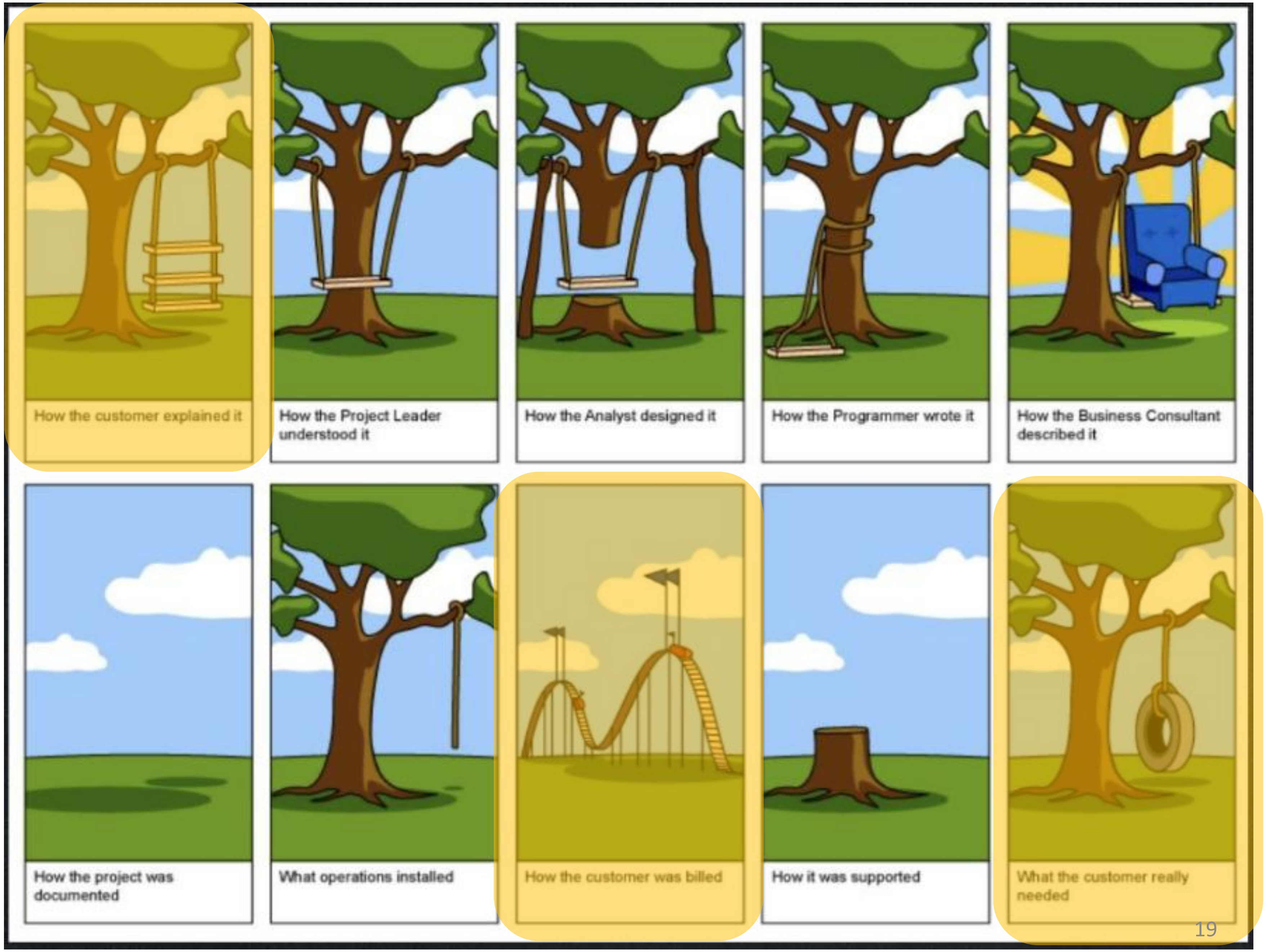


Know what to build

Understand the same way

The Communication Game

Customer:
I want a swing.


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

Requirement Analysis

- Requirement Analysis Phase is the formal phase during which customers and system analyst are
  - **Discussing**
  - **Brainstorming**
  - **Negotiating**
  - **Documenting the project requirements**

- To discover **problems and needs information** as much as possible to **protect errors and problems** that may occur later.

How do you understand your requirement?

- Let's say

Exam preparation is your requirement to accomplish.

Now, how much can you describe about what should be prepared?

How do you understand your requirement?

- When do you plan for the exam preparation?

  a) At the beginning of semester
  b) One month before the exam
  c) One week before the exam
  d) One day before the exam

WHY?

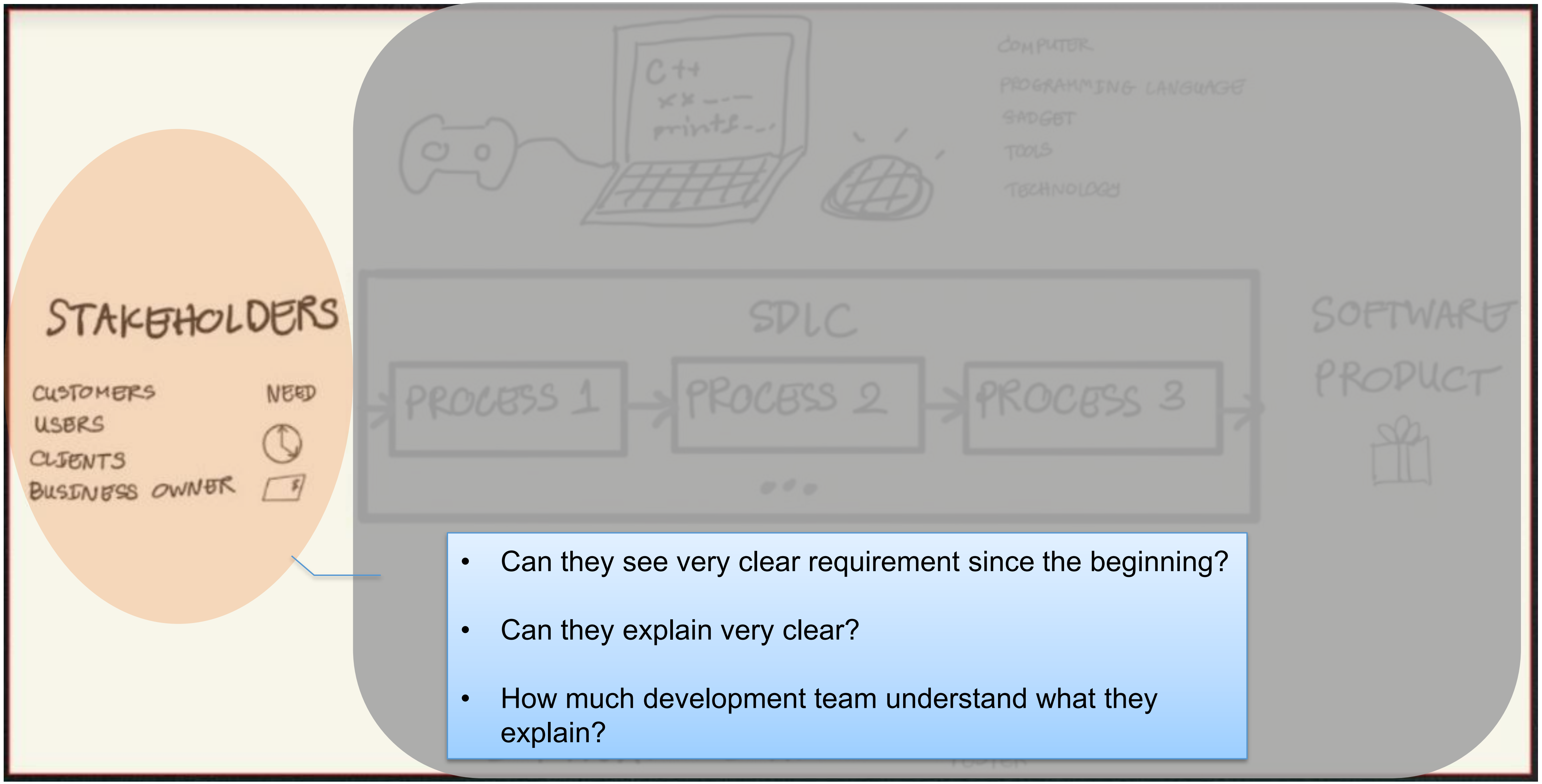
How do you understand your requirement?

- Can you keep doing as defined on your first plan?

  a) Yes

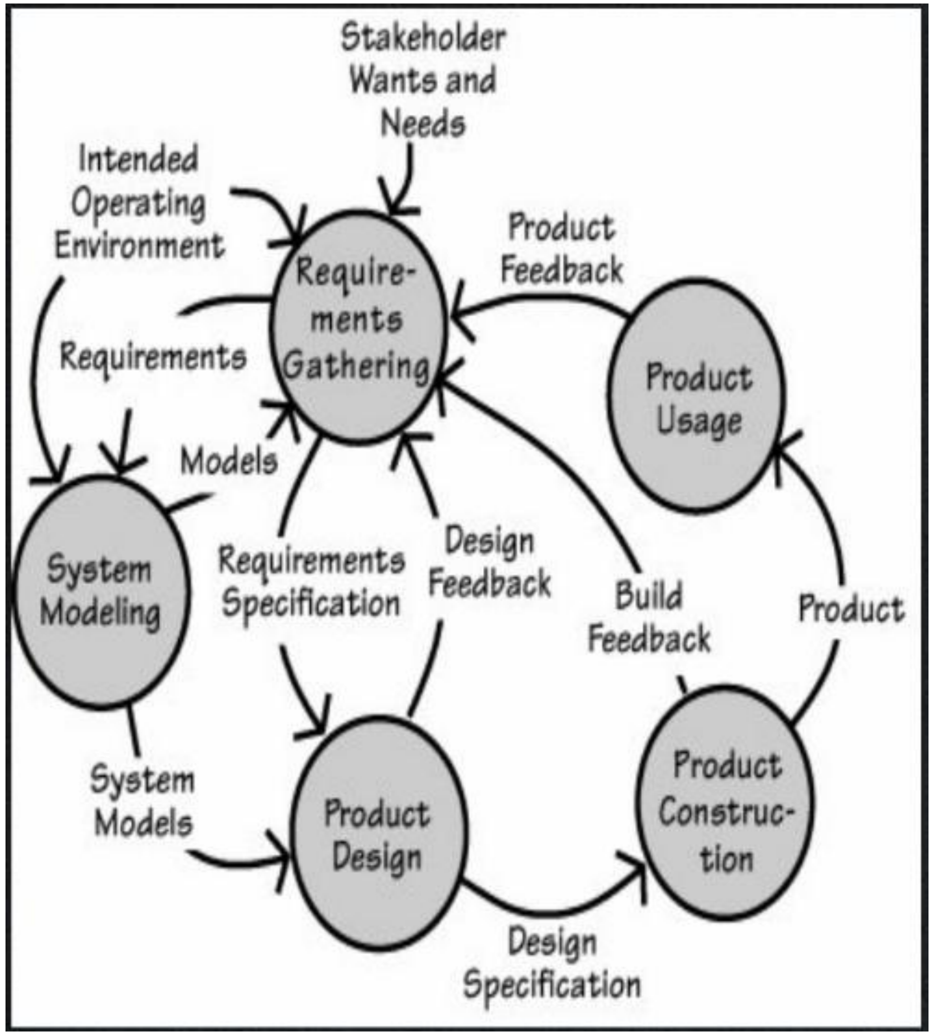
  b) No

WHY?



- Can they see very clear requirement since the beginning?
- Can they explain very clear?
- How much development team understand what they explain?

Requirement Analysis



Requirements process can be done **many times**.

- Each iteration produce some useful functionality.
- Once product is built and immediately begins to **evolve**. User demand more functionality.
- The product must be able to grow to accommodate the new command.
- Some may trigger new, previously unforeseen requirements.
- Some may change the delivered product.

We cannot control the evolution of the product,

requirement are not frozen at the moment that it is built.

It evolves over the period of time.

How can still make the project on budget and time ?

Goal of A Software Project

To develop quality software that meets customers real **needs**.

On **time**

On **budgets**

*Standish Group asked survey respondents to identify the most significant **factors** that contributed to project that were rated "**success**", "**late and did not meet expectations**" and "**fails**" respectively are related to requirements.*

(D. Leffingwell & D. Widrig, Managing S/w requirements: A unified Approach)

Goal of A Software Project

“late and did not meet expectations”

- Lack of **user involvements**
- **Incomplete** requirements and specification
- **Changing** requirements and specification

“fails”

- Unrealistic **Schedule & time** Frame
- Inadequate **staffing and resources**
- Inadequate technology **skills**.

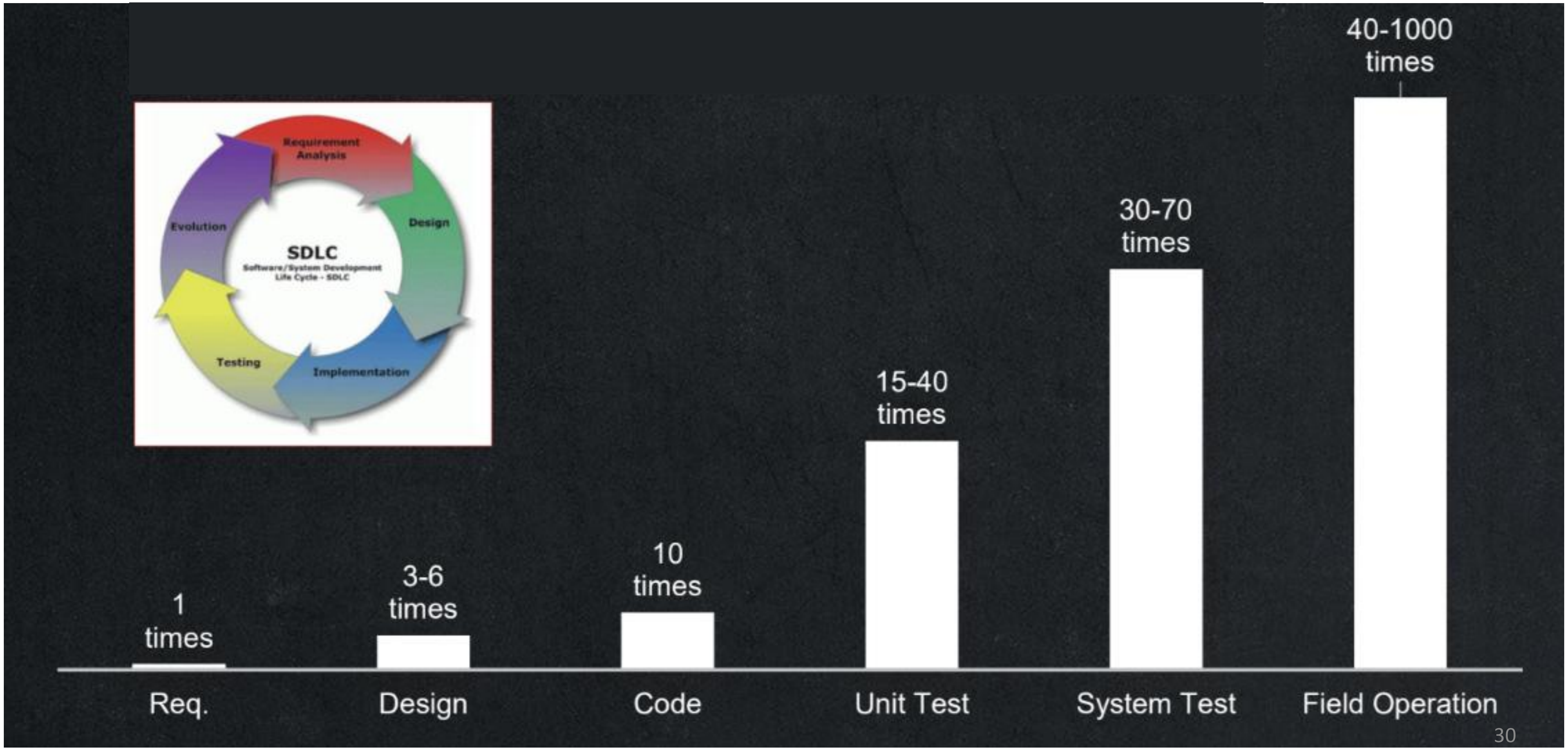
Goal of A Software Project

What are the primary "**success factors**" for the success of the project?

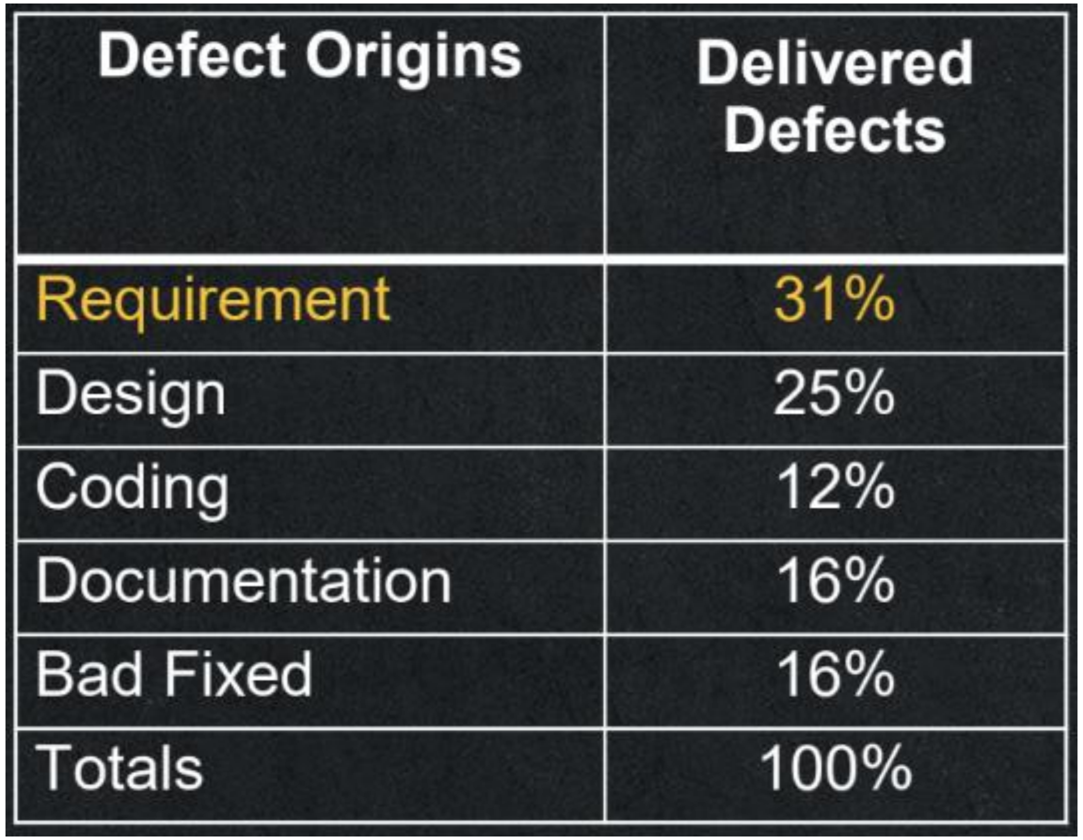
- **User involvement**
- Executive management support
- **Clear statement of requirements**

It seems clear that *requirements* deserve their place as a leading **root cause** of software problems and coding issue were a "non problem".

Relative Cost Correcting an Error



Relative Cost Correcting an Error



Defect Origins	Delivered Defects
Requirement	31%
Design	25%
Coding	12%
Documentation	16%
Bad Fixed	16%
Totals	100%

Capers Jones. "Managing of Software Requirements," 2002

https://www.developsense.com/blog/2014/10/facts-and-figures-in-software-engineering-research-part-2/

https://cs.nyu.edu/artg/Producing_Production_Quality_Software/Fall2005/lectures/SOFTWARE_QUALITY_IN_2002_CAPERS_JONES.pdf

“Obviously, it is ***better to correct requirements errors during requirements elicitation*** *than during design, code, test or post-deployment. Better still, a project team should minimize requirements errors as much as possible through the sound planning, elicitation, analysis, documentation, communication, verification and management of requirements.* ***Project success*** *depends on it.”*

Peter Gordon

Example cost repairing defects

- Re-specification
- Redesign (change requirement to spec -> redesign)
- Recoding (change specification to computer language ->

  Recoding -> retesting)

All are from the requirement change

- Code, Designing, Test cases are thrown away when they were based on the incorrect requirements.

Summary

- **What are Requirements?**
- **Why do we care about the requirements?**
  - To know what to build
  - To define what should be accomplished for users
  - **Developer and customers/stakeholders understand the <u>same way.</u>**
  - **Software project success factor**
- **Requirement Analysis >> evolving of requirements**
- **Goals of Software project >> requirements >> primary success factors**
  - Accomplished software
  - Time
  - Budget