

Orientation

SE-1221

**INTRODUCTION TO ENGINEERING DESIGN
(FOR SE)**

Richard Michael Coo

 [Repo](#) Slides:  [Online](#)  [PDF](#)



ABOUT ME

Richard Michael Coo

- Graduated BSSE 2007 at CPU
- Full-time Senior Node.js Developer at Dev Partners
- Part-time Instructor at CPU : SE dept.
- Specializes in:
 - Express.js
 - Nest.js
 - TypeScript
- Interested in:
 - Scala

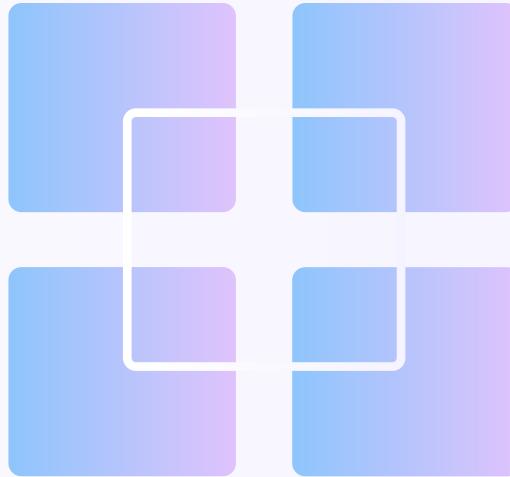
What is SE-1221?

Your first taste of the college motto:



Agenda

- Course Details
 - Schedule
 - Pre-requisite(s)
 - Mode of Delivery
- Expectations
- House Rules
- Meet and Greet
- Confusing Terminologies
 - Nature of SE (*or why not Engg 1011?*)
 - Software Engineering
 - Engineering Design
 - Design
- Ideation



The Last F2F IED Exhibit and Contest

DAYS BEFORE LOCKDOWN

The banner features the Central Philippine University logo at the top left, the College of Engineering logo at the top right, and a large purple banner in the center containing the event title. Below the banner, the theme is displayed in white text against a yellow background. At the bottom, several logos of engineering societies and departments are shown, along with the tagline "TURNING IDEAS INTO REALITY".

CENTRAL PHILIPPINE UNIVERSITY
COLLEGE OF ENGINEERING

COLLEGE OF ENGINEERING

17th FRESHMEN
INTRODUCTION TO ENGINEERING DESIGN
EXHIBIT

THEME:
"EMBRACING GLOBAL CHALLENGES
THROUGH SUSTAINABLE ENGINEERING DESIGN SOLUTIONS."

MARCH 9, 2020
ALUMNI PROMENADE CONCERT PARK

TURNING IDEAS INTO REALITY





Course Details

- Unlike Engg 1011: SE-1221 has a pre-requisite
 - thus, the different course code + course title
 - Pre-requisite: SE-1121 (*Software Development I*)
- Schedule
 - Lecture: **none** (`0` units lecture)
 - appears as TBA in SOS, same for Engg 1006
 - Laboratory: **14:00 - 17:00 S** (`3` units lab)
 - Mode of Delivery
 - F2F or online? *case-by-case basis*
 - group working sessions: agree per group
- No course has SE-1221 as a pre-requisite
 - but this course is designed for Freshmen

Expectations (1/2)

- Even if online:
 - 🧐 attend class hours from 2 to 5pm, and stay **focused**
 - 💻 work on tasks for group projects
- Everyone contributes:
 - 🔖 code: *since it's also your project for SE-1222 and SE-1223*
 - if coding is not your forte, in my class (**only**) - you can also contribute "other SE career paths stuff", e.g.:
 - 📋 tests - *mostly automated E2E/UI tests, maybe a little manual*
 - 🔍 requirements - validation, UX
 - what does not count as a contribution:
 - 📸 **only** contributing graphics, sprites (`BSSE != DMIA`, also plenty of assets online)
 - 🌐 "research" - in my experience, this usually means NOTHING

Expectations (2/2)

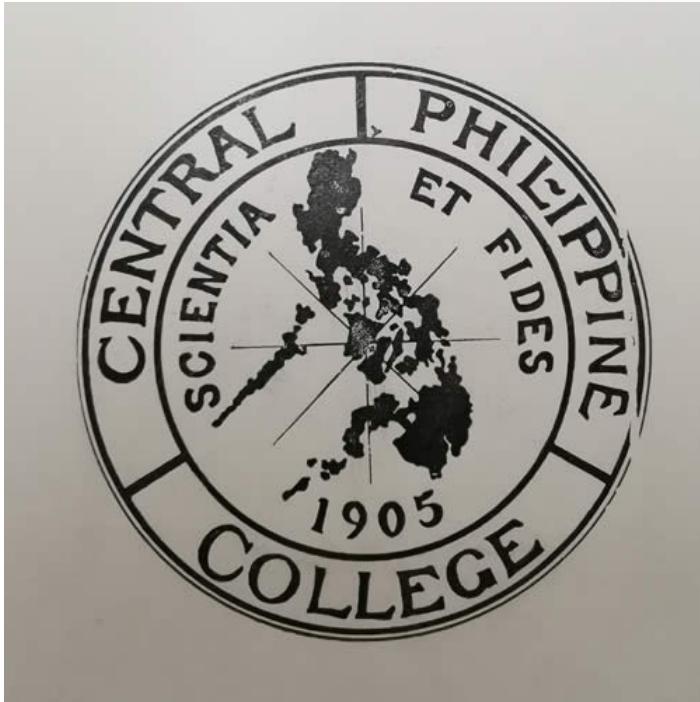
- JUL 17 don't cram: **DO some basic** project management
- work as a **group** of 5-6 people, no carries
- **MASTER** your data structures (*SE-1222*) and OOP (*SE-1223*)
- **Ask for help** from senior students or teachers
 - your knowledge is too lacking for a presentable product (*though sometimes there are exceptions*)

Not expected

- becoming expert Web app or game developer over a few months
 - Web app development (*SE-2125*) is next semester
 - Game development is an elective, and may not even be offered
 - Okay to be unable to "defend" Web-related or game-related code that are way to advanced for rookie learners
- 0 hours lecture: we'll have crash courses, but most of our 3-hour session is for WORK

House Rules (1/5)

VOLUNTEER TO PRAY BEFORE CLASSES



- Please pray from the heart, English not required:

12-13 As she kept on praying to the Lord, Eli observed her mouth. Hannah was **praying in her heart**, and her lips were moving but her voice was not heard. ¹⁹...the Lord remembered her plea,

-1 Samuel 1:12-13,19

- Please do not pray memorized prayers:

⁷ When you pray, don't babble on and on as the Gentiles do. They think their prayers are answered merely by repeating their words again and again. ⁹ **Pray like this:**

Our Father in heaven...

-Matthew 6:7,9

- Please avoid the *sign of the cross*. It's not Biblical 🤝

House Rules (2/5)

🚫 MUTE YOUR MIC WHEN NOT SPEAKING



House Rules (3/5)

PLEASE COMMUNICATE IN ENGLISH

- **English** is the medium of instruction
- Some of your classmates are **foreign students**
- **PRACTICE:** most software companies are multi-national

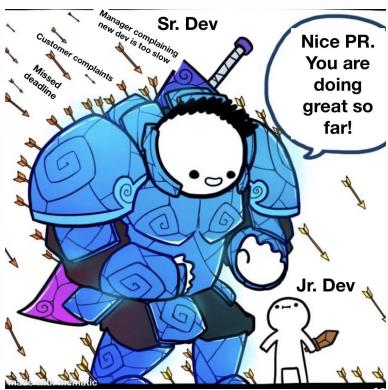


- it doesn't have to be perfect grammar, other countries more progressive but their English are not very good.
- always strive to recite, chat, ask questions, or help someone else in English (live class, Discord)
-  It's joke time!

House Rules (4/5)

DO GO THE EXTRA MILE

- fast learners will have lots of extra time
- don't be idle and waste time
- learn beyond what's required
 - make sure you *somewhat* understand them
 - exceptions - need 100%: `SE-1222`, `SE-1223`
- help your classmates in your spare time



Jesus taught us to walk the extra mile

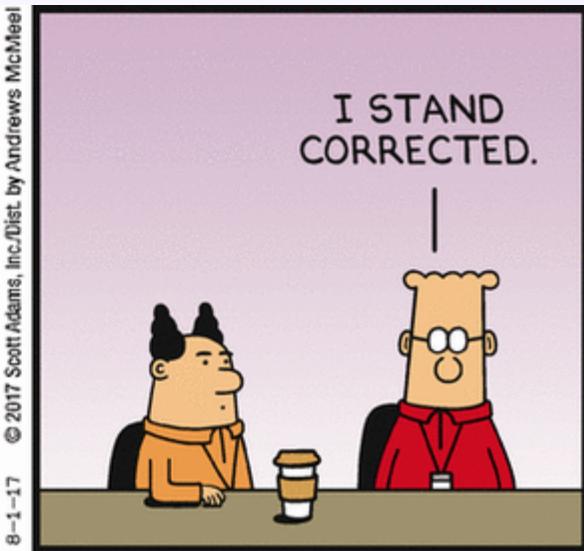
**If anyone forces you to go one mile,
go with them two miles.**

-Matthew 5:41

House Rules (5/5)

THERE ARE NO STUPID QUESTIONS

- don't be shy, you're all rookies
- students that ask questions are the good ones who usually get `1.0`



A white puppy with a blue collar is looking up. A white speech bubble above it contains the word "Hello".

Hello

Meet and Greet

I don't know anyone, first time handling this batch

- by group
- turn on cam
- you can:
 - introduce individually
 - or have the leader introduce each member
- Tell us:
 - team name
 - for each member: **name + nickname**
 - one interesting/funny fact about the group 😂

Nature of SE

- complex
- intangible
 - not governed by natural sciences like physics, chemistry, etc.
 - computer science as the foundation
 - hard to understand development effort
- labor-intensive
 - the bulk of the costs is the salary of the software development team
 - in other engineering disciplines - material and manufacturing takes up a huge %
 - hard to automate (*still true as of 2023*)
- Quality of software engineers vary
- Perceived ease of modification
 - changes usually introduce unforeseen defects (a.k.a. ~~de~~ bugs)

Additional resources: [Slides from University of Ottawa, Nature of Software from SEBOK Wiki](#)

Software engineer

- when used to denote a profession: is much more broadly employed than “software engineering” as an academic discipline or a degree program.
- There are many **more individuals** with a job title or professional identity of a “software engineer”
 - than those who have graduated from BSSE programs.
- More confusion: software engineering is a course/subject in BSCS and sometimes, BSIT or BSCpE programs

See [ACM CC2020 page 28](#)

As much as I want to promote the department that molded me, but 1 Peter 3:10 teaches us to speak honestly:



YOU DON'T NEED TO BE A BSSE GRADUATE TO BECOME A "SOFTWARE ENGINEER"!



You may not end up with a job title of "software engineer" (or equivalent) even with a BSSE degree, esp. in less "technical" roles

- but maybe: *business analyst*, "QA Engineer" (*popular but inaccurate*), *Test Engineer*, *ScrumMaster*, etc.

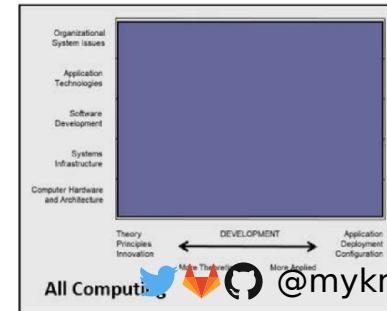
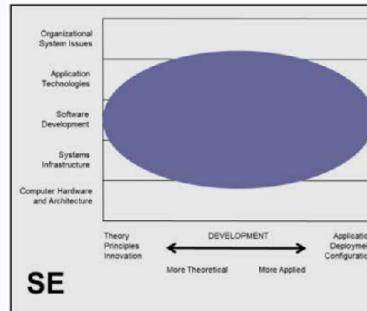
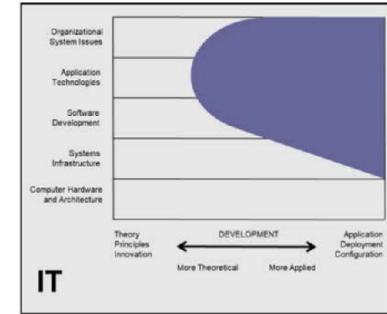
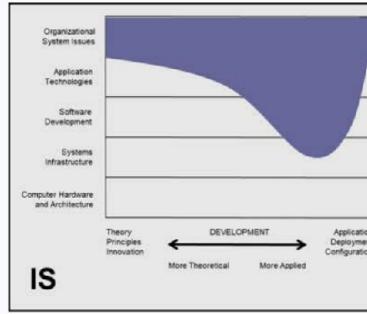
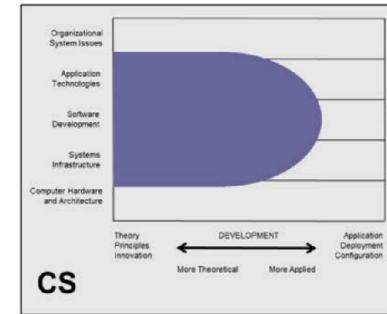
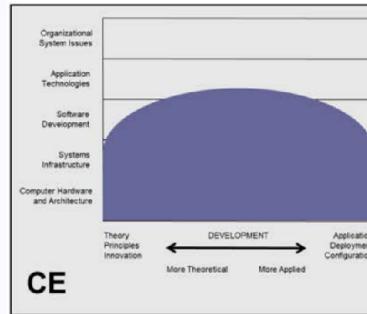
Software engineering

- would usually refer to the discipline
 - body of knowledge (SWEBOK): 15 knowledge areas
 - SE focuses on creating software-based solutions, but it is **much more than programming.**
- SE emphasizes the use:
 - of appropriate software development practices,
 - the integration of engineering rigor, along with
 - the ability to apply advanced algorithms and data structures developed in computer science.
- strongly focused on designing software systems that are:
 - high-quality
 - meets or exceeds customer expectations



manages complexity and change

See ACM CC2020 page 29



@myknbani

O'REILLY®

Software Engineering at Google

Lessons Learned
from Programming
Over Time



Curated by Titus Winters,
Tom Mansreck & Hyrum Wright

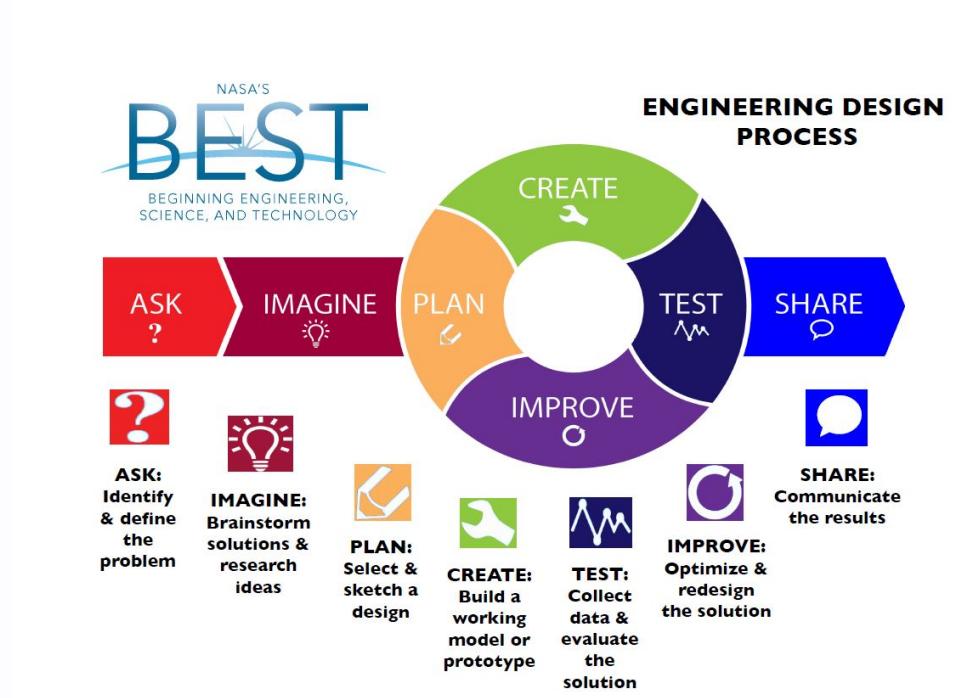
This book is a must have! Recommended for students!

Software Engineering

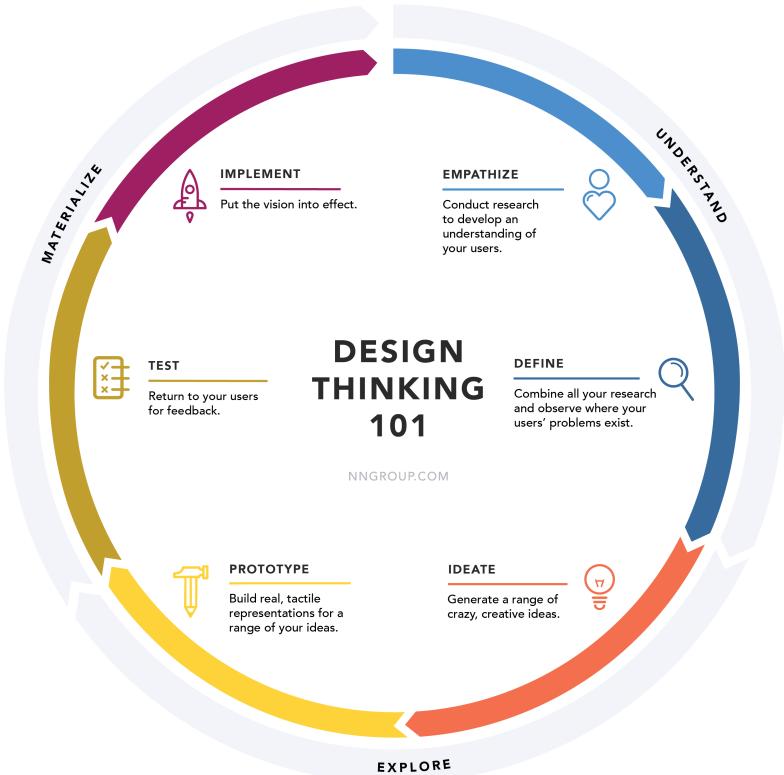
- Boring but still accurate definition:
is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software.
- Simple, modern, and fresh:
Programming over time.
 - There's a lot to unpack from those three words.
 - **Time** introduces two of SE's mortal enemies:
 - complexity
 - change
 - Software engineers program! They **write code!**
 - civil eng'g: not always a good analogy to SE

DESIGN: in *engineering design*

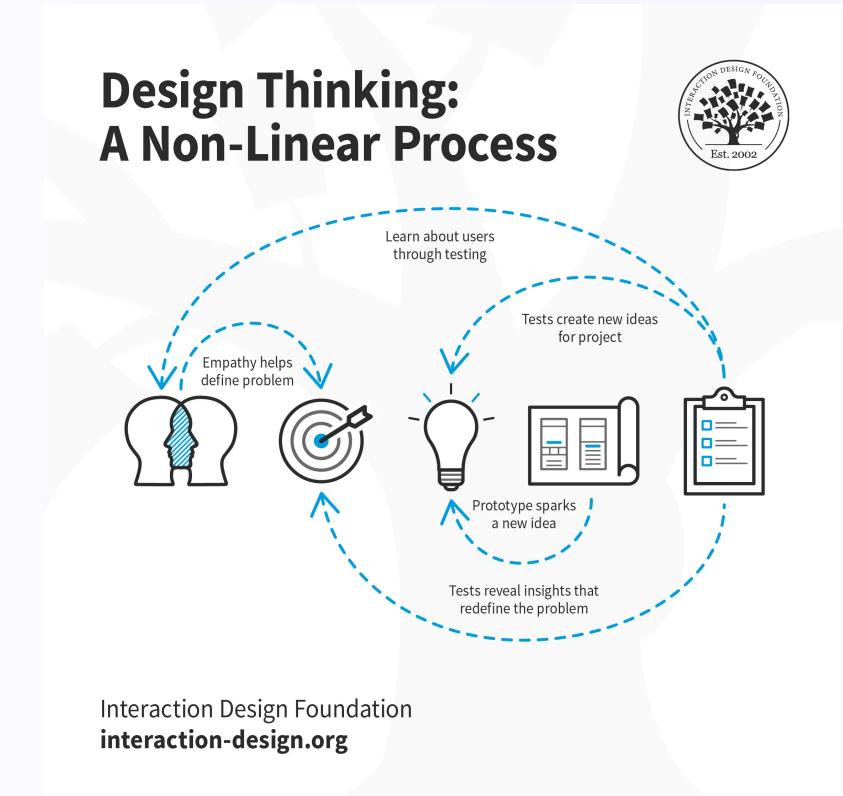
- is an iterative process used to identify problems and develop and improve solutions.



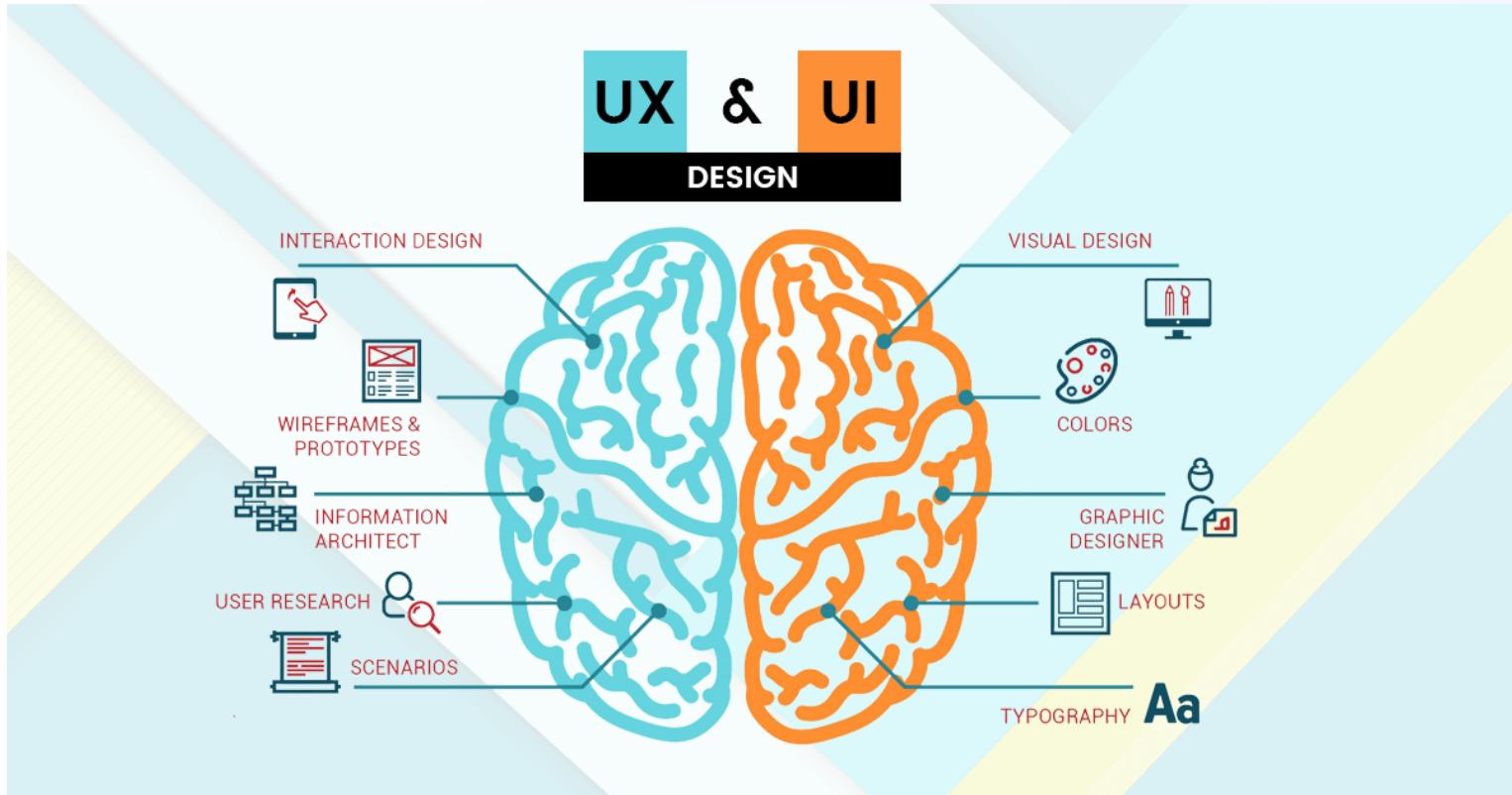
DESIGN: in *design thinking*



- a hands-on, user-centric approach to problem solving can lead to innovation.

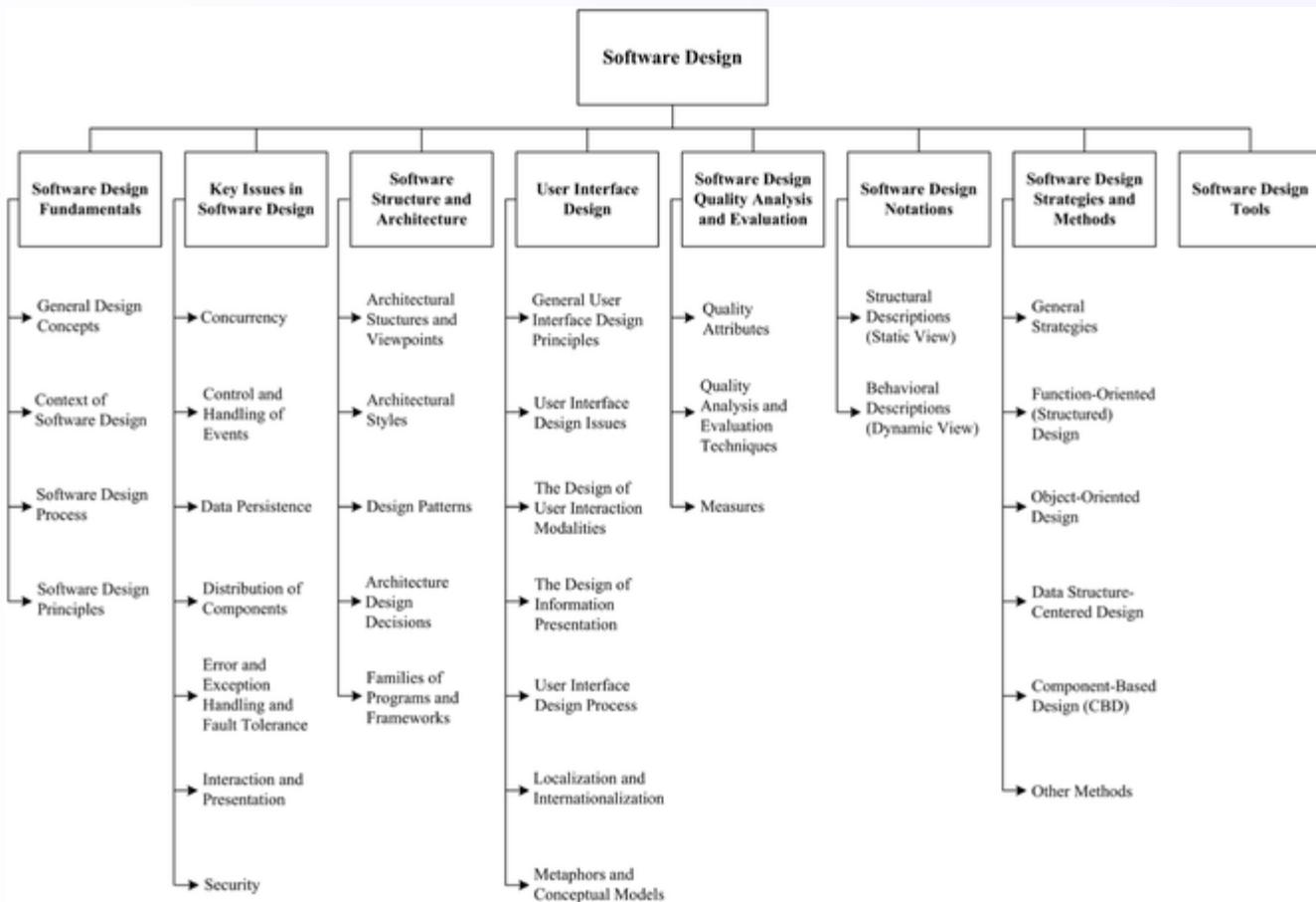


DESIGN: in *user interface design*



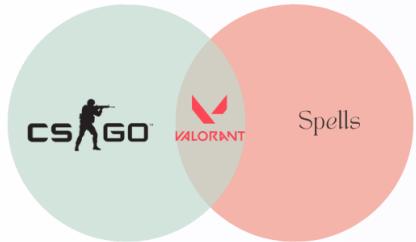
DESIGN: in the *software design* KA of the SWEBOK

- is the software engineering life cycle activity in which **software requirements** are analyzed in order to
- produce a **description of the software's internal structure** that will serve as the **basis for its construction**.
 - architecture: "*high-level design*" (SE 2223: Software Architecture)
 - detailed design: "*low-level design*" (incorporated across different courses), e.g.:
 - OOP design principles will be dicussed in SE-1223 (*Software Development II*)
 - Design patterns will be discussed in SE-2122 (*Software Component Design*)
- Many ways to write code for the same problem:
 - "well-designed" code allow us to navigate complexity
 - it allow us to introduce changes without breaking something else
- Your code **always has a design** (it always has an *internal structure*) whether it's documented or not.
 - Design is not always "*sketching diagrams*" akin to blueprints
 - There's a design technique called test-driven development (TDD) that does this – no diagrams!
 - Designs can be as simple as a rough sketch on a whiteboard that is finished in code

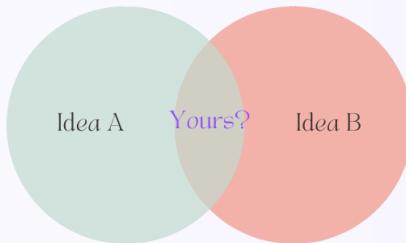
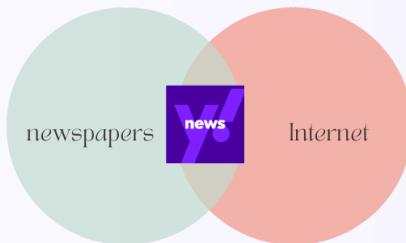


Ideation (1/4)

MASHUP TECHNIQUE



- repurpose an existing product for better fit
- combine existing ideas



Ideation (2/4)

- You are the first customer of your product
- I hate _____

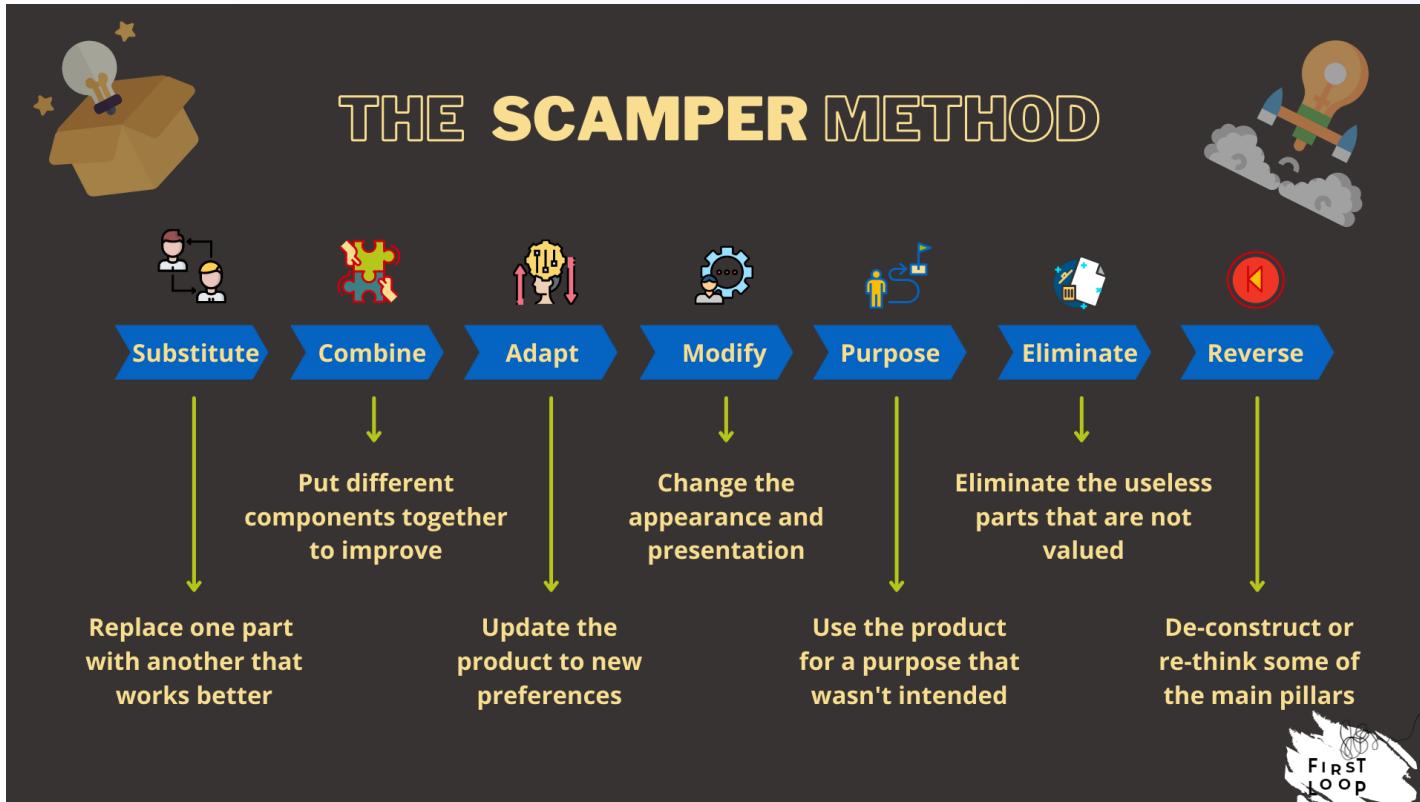
 Commuting? Scammer Taxi drivers in Manila?



- these products are usually innovative and **disruptive**
 - If launched successfully, the one you hated will now hate you
-  I hate banks



Ideation (3/4): SCAMPER



Ideation (4/4): Brainstorming



Rules of Brainstorming



Defer Judgment



One Conversation at a Time



Encourage Wild Ideas



Be Visual



Build on the Ideas of Others



Go for Quantity



Stay Focused on the Topic

© IDEO 2020

Credits

- Berkeley Method of Entrepreneurship: Opportunity Identification (*Ikhlaq Sidhu*)
- others: TODO