SOFTWARE ENGINEERING

Class 2, Course Overview and Introduction

TOPICS

- Course Overview.
- Foundation.
- Reference materials.

COURSE OVERVIEW

GOAL

Learn software engineering concepts by building a highly scalable application deployed to the cloud.

Theory

Mid Sem

Project

12-14 Classes

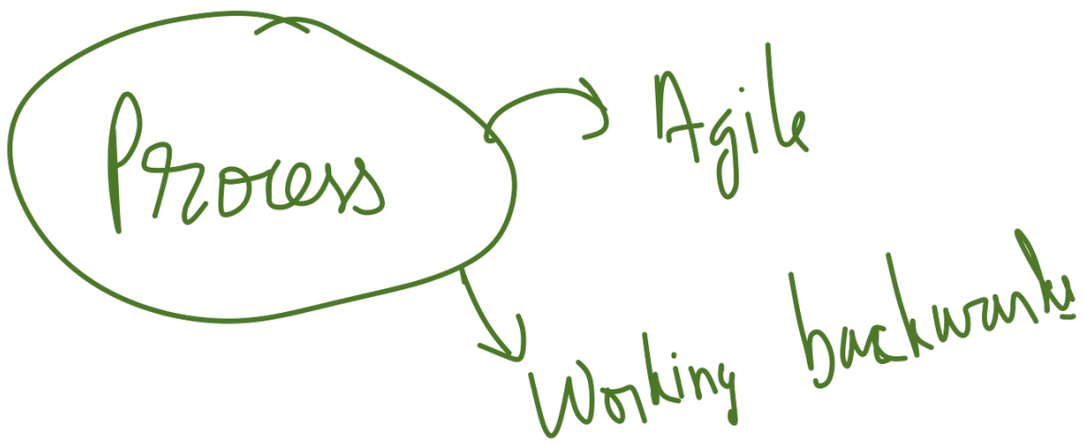
(Software Engineering Basics)

12-14 Classes

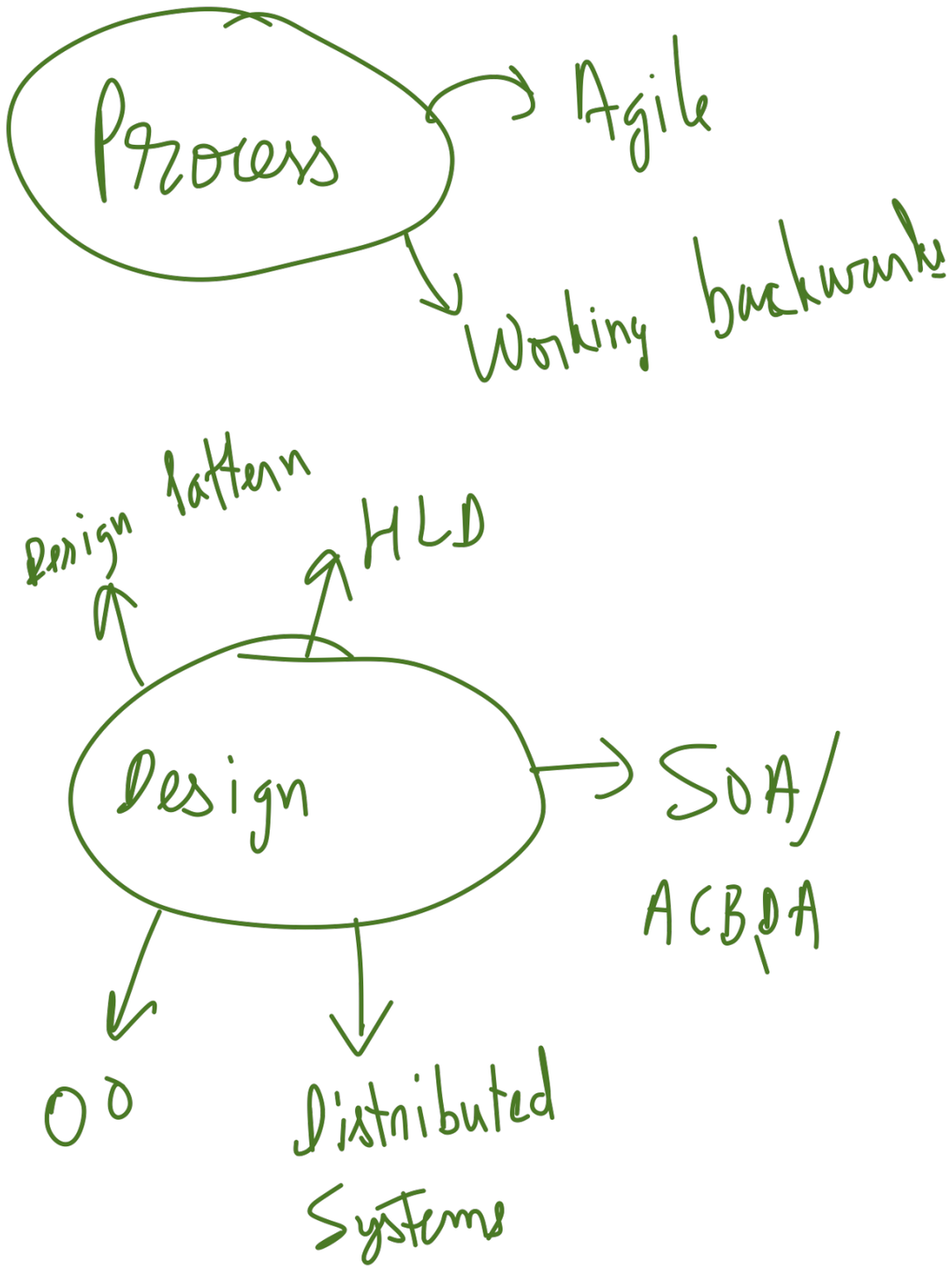
(Reviews, Demo & Guest lecture)

2-4 Optional Classes for learning Java/ Cloud Syste

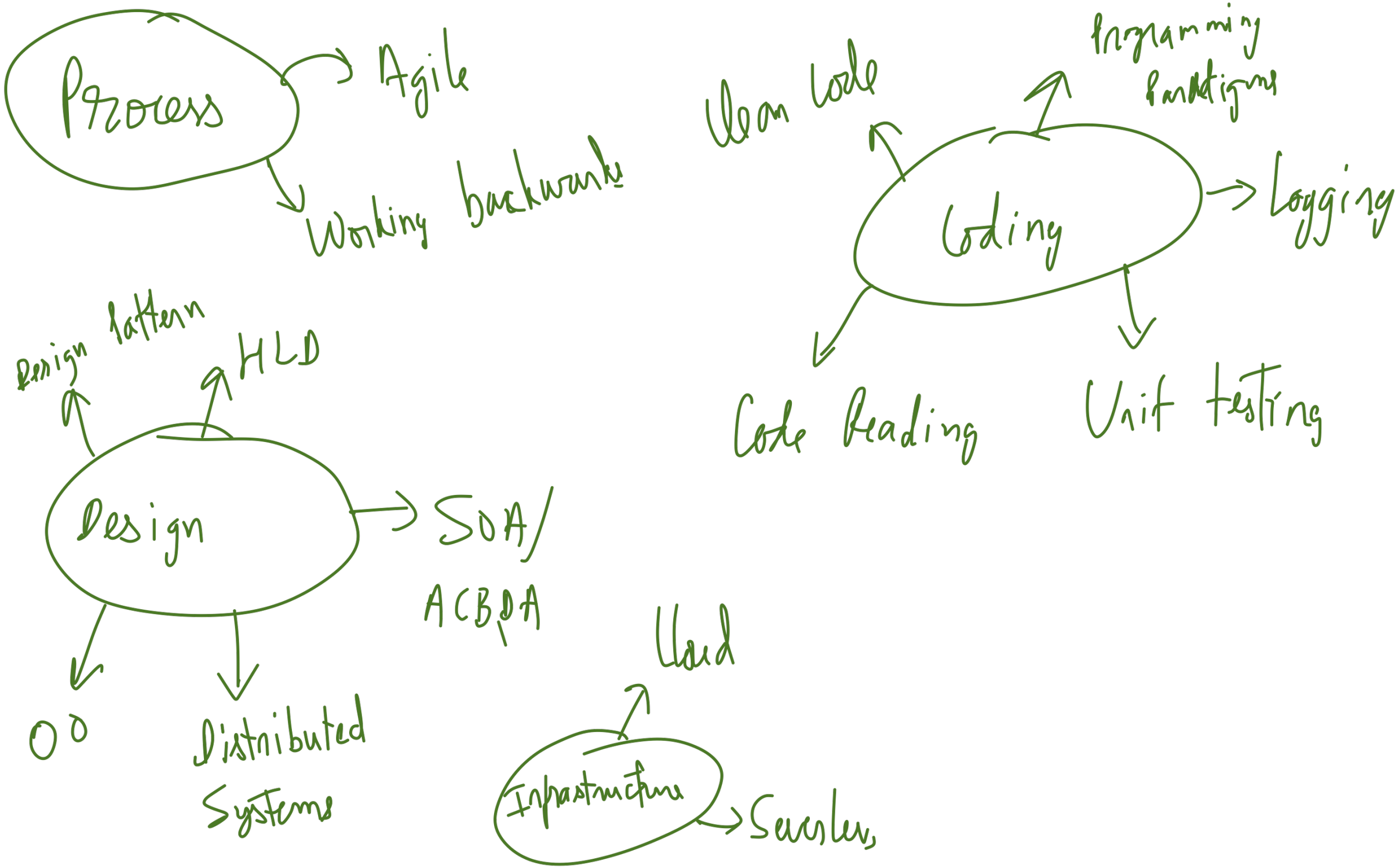
THEORY



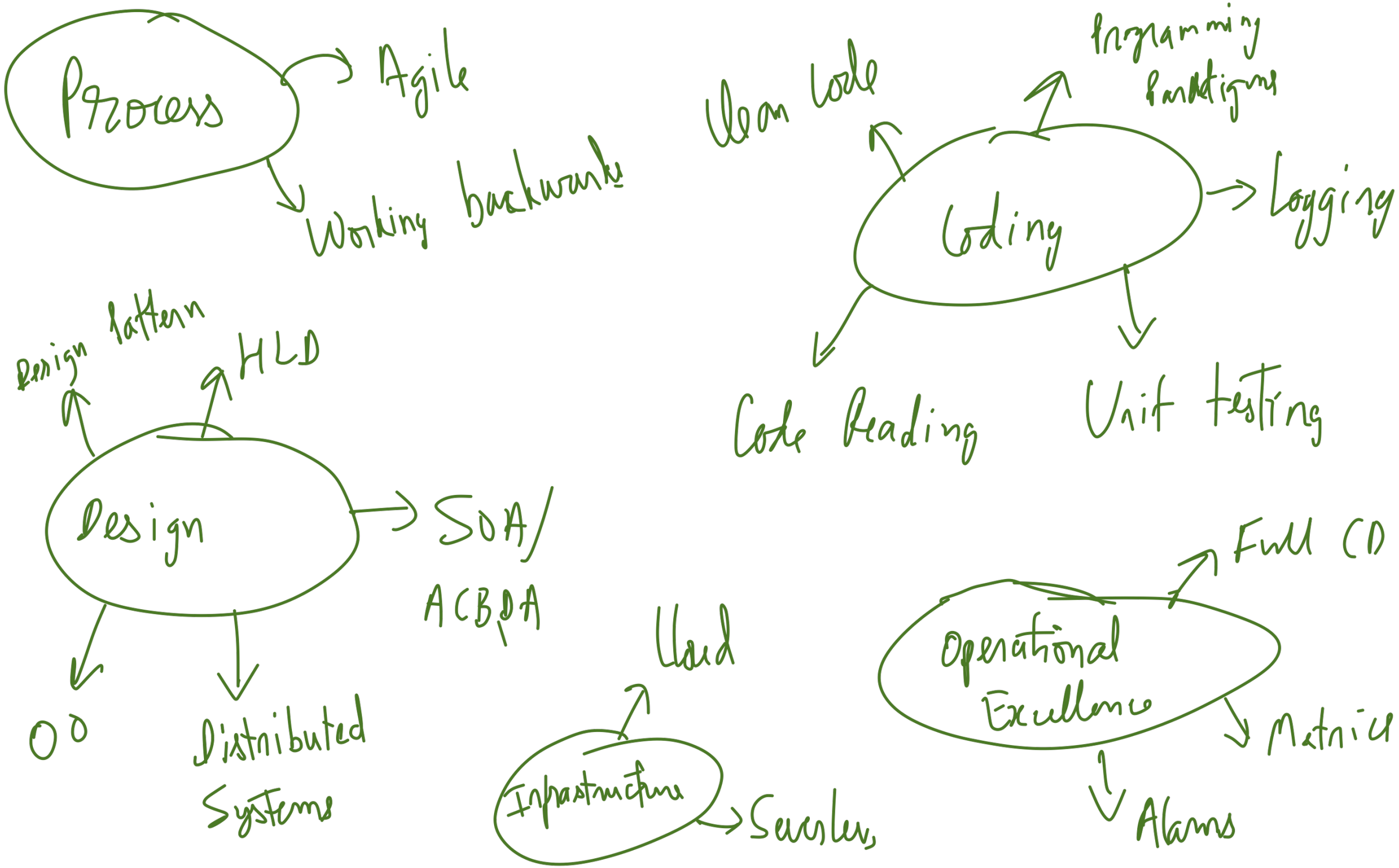
THEORY


Process
Agile
Working backwards
Design pattern
HLD
Design
SOA/
ACBDA
OO
Distributed
Systems

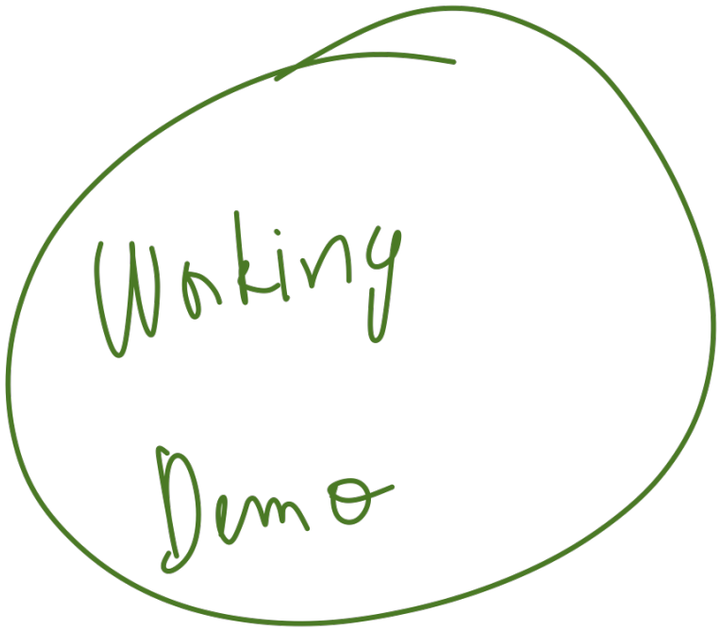
THEORY


Process
Agile
Working backwards
Design pattern
HLD
Design
SOA/
ACBDA
OO
Distributed
Systems
Clean code
Programming
Paradigms
Coding
Logging
Code Reading
Unit testing
Cloud
Infrastructure
Security

THEORY


Process
Agile
Working backwards
Design pattern
HLD
Design
SOA/
ACBDA
OO
Distributed
Systems
Clean code
Programming
Paradigms
Coding
Logging
Code Reading
Unit testing
Cloud
Infrastructure
Security
Full CD
Operational
Excellence
Metrics
Alarms

PROJECT [Team Size : 2 - 6]



Type of Evaluation	Weightage (in %)
Mid SemExam	**20%**
PRFAQ Review (In Class)	**10%**
Design Review (In Class)	**10%**
Final Project Demo (In Class)	**40%**
Class interaction/Quiz	**20%**

20%

80%

No text book / all reference materials available online

FOUNDATION

LEARNING.
EXPLORATION.
DISCOVERY.

“

“the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.”

“

“the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.”

engineering

/ɛnʤɪˈnɪərɪŋ/

noun

1. the branch of science and technology concerned with the design, building, and use of engines, machines, and structures.
2. the action of working artfully to bring something about.
   "if not for his shrewd engineering, the election would have been lost"

Translations, word origin, and more definitions

ENGINEERING

- Requirements
- Design
- Construction
- Testing
- Maintenance


HOW DO THEY KNOW THE LOAD LIMIT ON BRIDGES, DAD?
LOAD LIMIT 10 TONS


THEY DRIVE BIGGER AND BIGGER TRUCKS OVER THE BRIDGE UNTIL IT BREAKS.


THEN THEY WEIGH THE LAST TRUCK AND REBUILD THE BRIDGE.
WATTERSON


OH. I SHOULD'VE GUESSED.
DEAR, IF YOU DON'T KNOW THE ANSWER, JUST TELL HIM!

https://www.youtube.com/watch?v=7QCkK_p6TZA


Clojure
Ruby
Scala
python
Java
Microsoft
.net
ERLANG

> The conversion of an idea to an artefact, which engages both the designer and the maker, is a complex and subtle process that will always be far closer to art than science.

-Eugene S. Ferguson,
Engineering and the Mind's eye

SOFTWARE ENGINEERING

- Requirements
- Design
- Construction
- Testing
- Maintenance

“

Walking on water and developing software from a specification are easy if both are frozen

-Edward V Berard

“

Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away

-Antoine de Saint-Exupery

> Let us change our traditional attitude to the construction of programs. Instead of imagining that our main task is to instruct a computer what to to, let us concentrate rather on explaining to human beings what we want a computer to do.
>
> *-Donald Knuth*

https://github.com/openssl/openssl/blob/20b82b514d81a64f5b240788e5051167456af379/ssl/d1_both.c#L1326

Program testing can be used to show the presence of bugs, but never to show their absence!

-Edsger Dijkstra

> The fundamental problem with program maintenance is that fixing a defect has a substantial (20-50 percent) chance of introducing another. So the whole process is two steps forward and one step back..
>
> *-Fred Brooks*

NO SILVER BULLET

- 0 μm – 1971
- 6 μm – 1974
- 3 μm – 1977
- 1.5 μm – 1981
- 1 μm – 1984
- 800 nm – 1987
- 600 nm – 1990
- 350 nm – 1993
- 250 nm – 1996
- 180 nm – 1999
- 130 nm – 2001
- 90 nm – 2003
- 65 nm – 2005
- 45 nm – 2007
- 32 nm – 2009
- 22 nm – 2012
- 14 nm – 2014
- 10 nm – 2016
- 7 nm – 2018
- 5 nm – 2020

Future

- 3 nm – *~2021*
- 2 nm – *~2024*

Fred Brooks in 1986

THE EMPEROR'S OLD CLOTHES

The 1980 ACM Turing Award Lecture
Charles Antony Richard Hoare

EXERCISE

REFERENCES/SOURCES

- https://www.infoq.com/presentations/Software-Engineering/ by Glenn Vanderburg
- https://www.infoq.com/podcasts/taking-back-software-engineering/
- “No Silver Bullet - Essence and Accidents of Software Engineering” by Frederick P. Brooks. Available at IEEExplore and https://en.wikipedia.org/wiki/No_Silver_Bullet
- SWEBOK v3.0 - https://www.computer.org/education/bodies-of-knowledge/software-engineering
- “The Emperor's Old Clothes” - https://dl.acm.org/doi/10.1145/358549.358561
- https://www.comp.nus.edu.sg/~damithch/pages/SE-quotes.htm?type=maintenanceQuotes
- https://en.wikipedia.org/wiki/Moore%27s_law