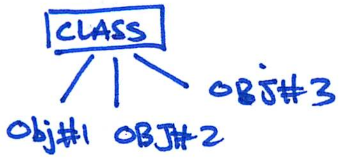
CMPE 202

COURSE NOTES

PART I

PAUL NGUYEN

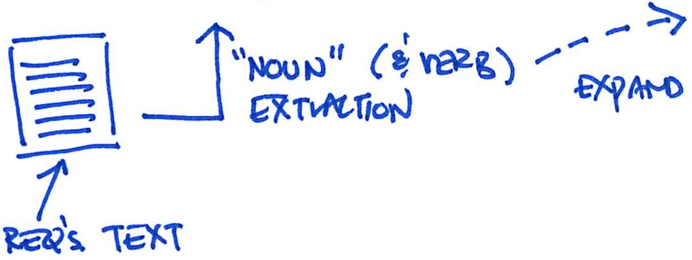
OO CONCEPTS



- class defn what objects "KNOW" & "DO"
- OBJECTS are "INSTANCES" OF A CLASS

CLASSICAL DATA MODELING

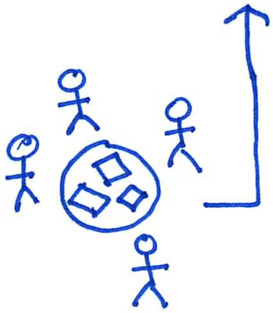
- ENTITY
- RELATIONSHIP



Object-ORIENTED MODELING

- CLASS
- RESPONSIBILITIES
- COLLABORATORS

} CRC

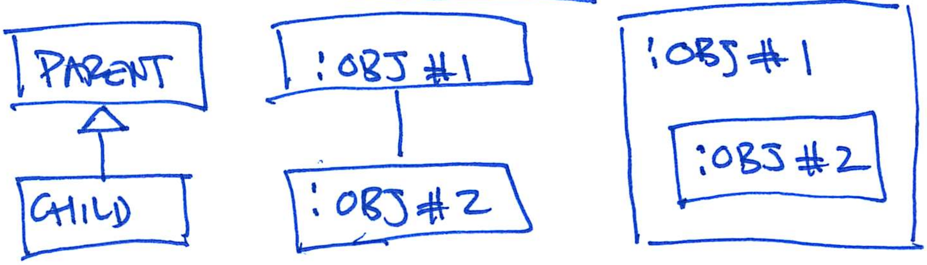


DISCOVER OBJECTS By Acting out SCENARIOS

ERD's

1-to-MANY
1-to-ONE
optionality
DEPENDENCY

OBJECT RELATIONSHIPS



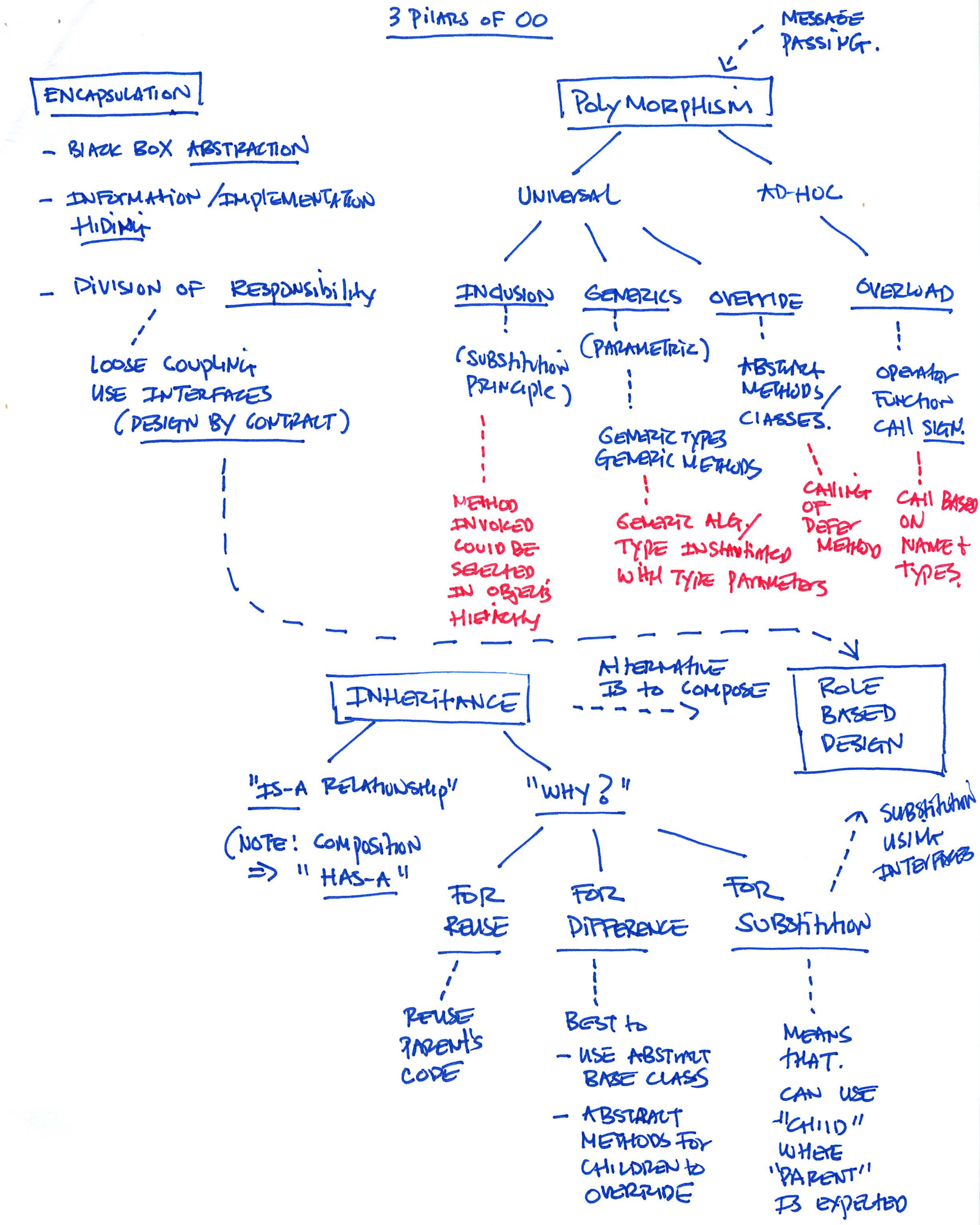
"CLASS-LEVEL" — SPECIALIZATION

"FRIENDS" — COLLABORATION

"WHOLE - PART" — COMPOSITION

SPECIAL CASE ?

3 PILARS OF OO



MESSAGE PASSING.

ENCAPSULATION

- BLACK BOX ABSTRACTION
- INFORMATION/IMPLEMENTATION HIDING
- DIVISION OF RESPONSIBILITY

LOOSE COUPLING
USE INTERFACES
(DESIGN BY CONTRACT)

POLYMORPHISM

UNIVERSAL

INCLUSION
(SUBSTITUTION PRINCIPLE)
METHOD INVOKED COULD BE SELECTED IN OBJECTS HIERARCHY

GENERICS
(PARAMETRIC)
GENERIC TYPES
GENERIC METHODS
GENERIC ALG/TYPE INSTANTIATED WITH TYPE PARAMETERS

OVERRIDE
ABSTRACT METHODS/CLASSES.
CALLING OF DEFER METHOD

AD-HOC

OVERLOAD
OPERATOR FUNCTION CALL SIGN.
CALL BASED ON NAME + TYPES.

INHERITANCE

ALTERNATIVE IS TO COMPOSE

ROLE BASED DESIGN

"IS-A RELATIONSHIP"

(NOTE: COMPOSITION ⇒ "HAS-A"

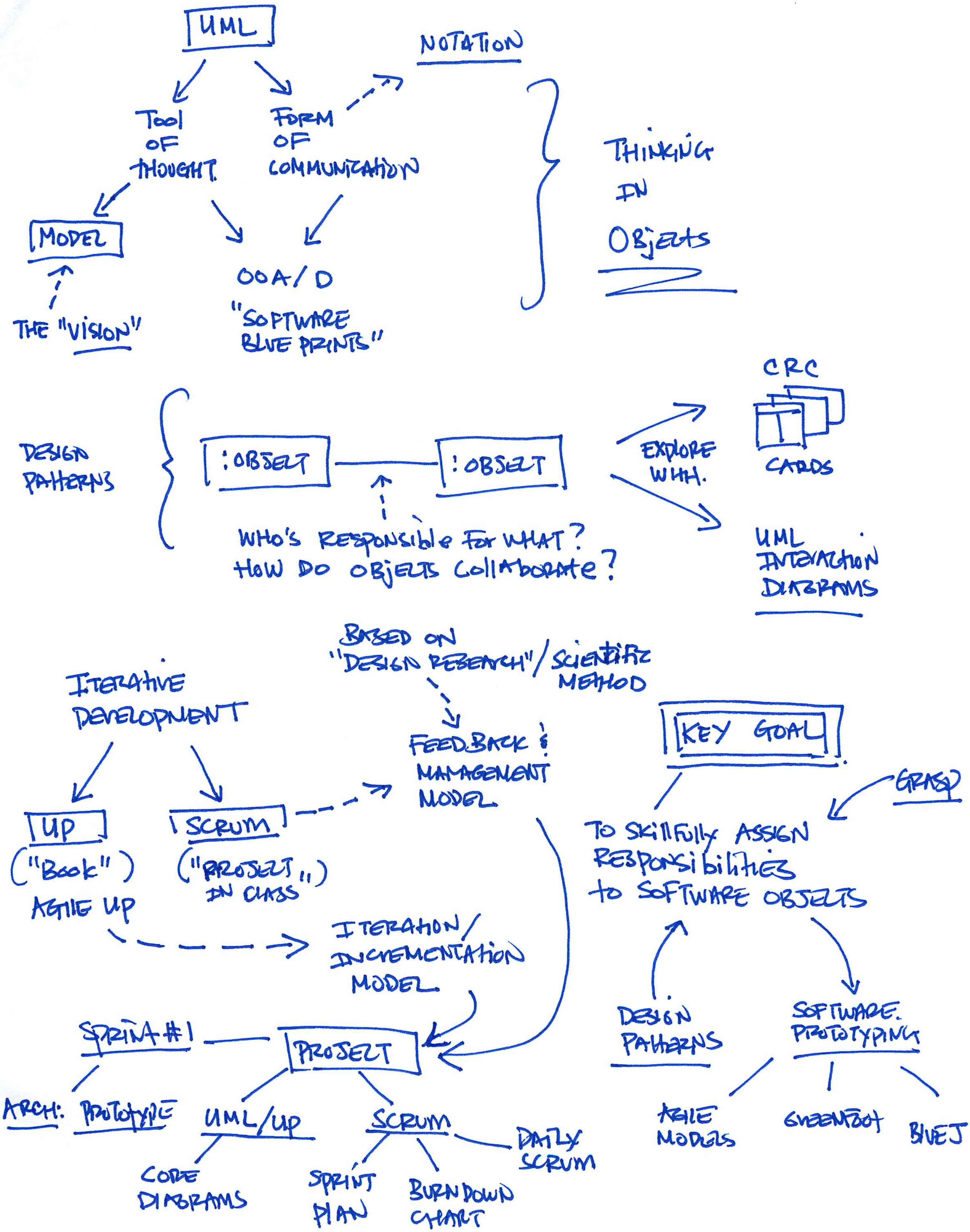
"WHY?"

FOR REUSE
REUSE PARENT'S CODE

FOR DIFFERENCE
BEST TO
- USE ABSTRACT BASE CLASS
- ABSTRACT METHODS FOR CHILDREN TO OVERRIDE

FOR SUBSTITUTION
MEANS THAT. CAN USE "CHILD" WHERE "PARENT" IS EXPECTED

SUBSTITUTION USING INTERFACES


UML
NOTATION
Tool of thought
FORM OF COMMUNICATION
THINKING IN Objects
MODEL
OOA/D
THE "VISION"
"SOFTWARE BLUE PRINTS"
CRC
CARDS
DESIGN PATTERNS
:OBJECT
:OBJECT
EXPLORE WITH.
WHO'S RESPONSIBLE FOR WHAT?
HOW DO OBJECTS COLLABORATE?
UML INTERACTION DIAGRAMS
BASED ON "DESIGN RESEARCH"/SCIENTIFIC METHOD
ITERATIVE DEVELOPMENT
FEEDBACK & MANAGEMENT MODEL
KEY GOAL
GRASP
TO SKILLFULLY ASSIGN RESPONSIBILITIES TO SOFTWARE OBJECTS
UP
("Book")
AGILE UP
SCRUM
("PROJECT IN CLASS")
ITERATION/INCREMENTATION MODEL
DESIGN PATTERNS
SOFTWARE PROTOTYPING
SPRINT #1
PROJECT
ARCH: PROTOTYPE
UML/UP
SCRUM
DAILY SCRUM
CODE DIAGRAMS
SPRINT PLAN
BURNDOWN CHART
AGILE MODELS
GREENFOOT
BLUEJ

202-3

ANALYSIS

- REQUIREMENTS ANALYSIS --> INVESTIGATE REQUIREMENTS
- OBJECT-ORIENTED ANALYSIS --> INVESTIGATE OBJECTS
  "DOMAIN OBJECTS"

} FIND & DESCRIBE THINGS !
"IN PROBLEM SPACE"

"DO THE RIGHT THING"
"DO THE THING RIGHT"
} WHAT VS. HOW (CLASSICAL DEFN')

DESIGN

- OBJECT-ORIENTED DESIGN --> CONCEPTUAL SOLUTION (NOT BOUND TO PROG. LANGUAGE)

DEFINE HOW OBJECTS (IN SOLUTION SPACE) WORK TOGETHER (COLLABORATE) TO FULFILL REQUIREMENTS.

Book's sample workflow

1. DEFINE USE CASES
2. DEFINE DOMAIN MODEL
3. DEFINE INTERACTION DIAGRAM
4. DEFINE DESIGN CLASS DIAGRAM.

CLASS

- CONCEPTUAL CLASS ---> ANALYSIS CLASS / DOMAIN MODEL
  "REAL WORLD CONCEPT"
- SOFTWARE CLASS ---> "SPEC. FOR CODE" / LOGICAL DESIGN
- IMPLEMENTATION CLASS ---> "LANG. SPECIFIC" / PHYSICAL DESIGN

ITERATIVE / EVOLUTIONARY DEVELOPMENT — USES FEEDBACK to "EVOLVE" SPECIFICATIONS

- EARLY PROG. & TESTING OF PARTIAL SYSTEM
- DEVELOPMENT STARTS BEFORE REQ'S. FULLY DEF'N.

COMMON KNOWLEDGE FOR UML PRACTITIONERS

- UP — UNIFIED PROCESS — (1999 JBR) ← THE THREE AMIGOS
- RUP — RATIONAL UNIFIED PROCESS

KEY IDEAS FROM OTHER AGILE PRACTICES.

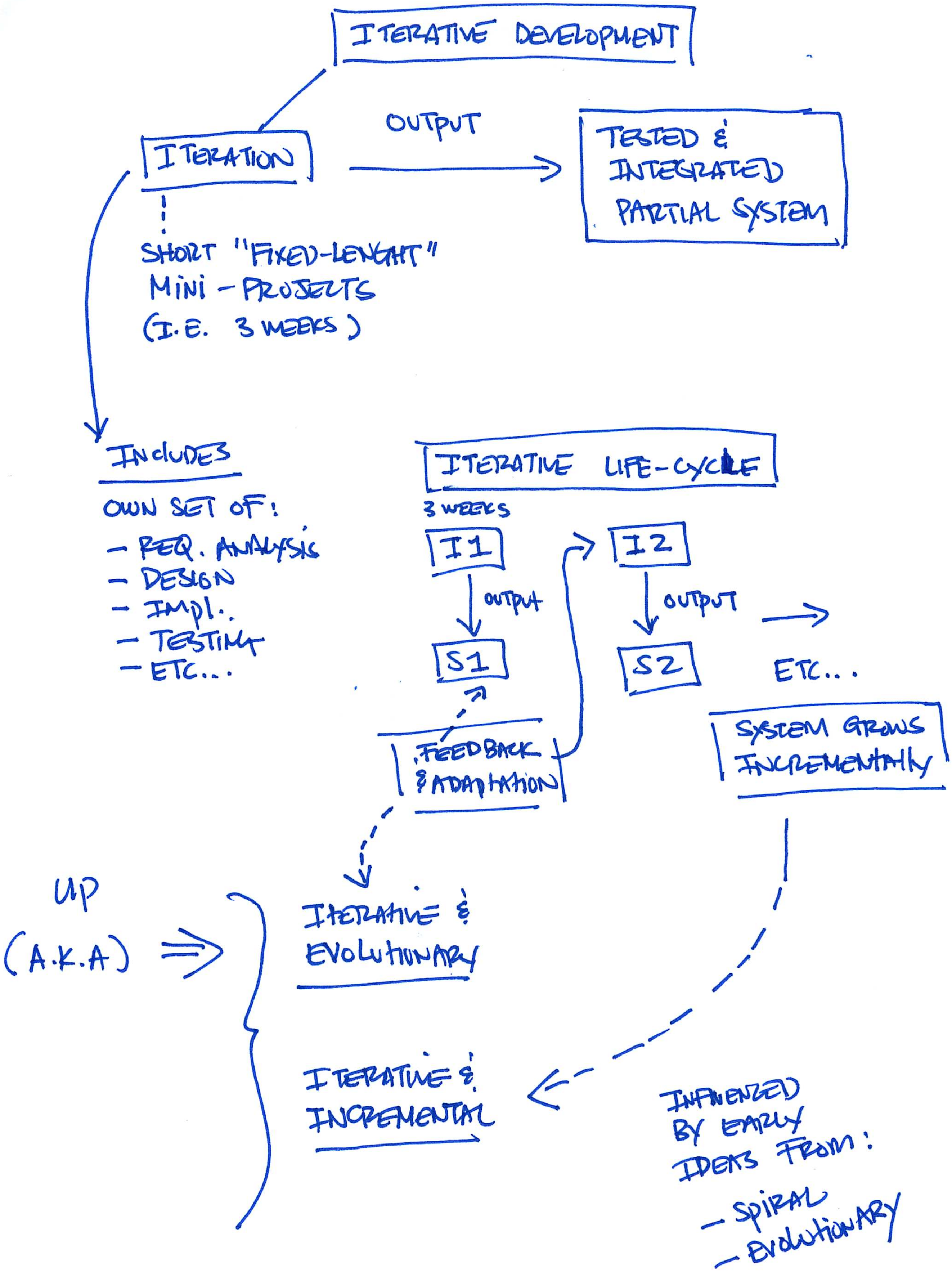
XP
- TDD (TEST-DRIVEN DEVELOPMENT)
- REFACTORING
- CONTINUOUS INTEGRATION

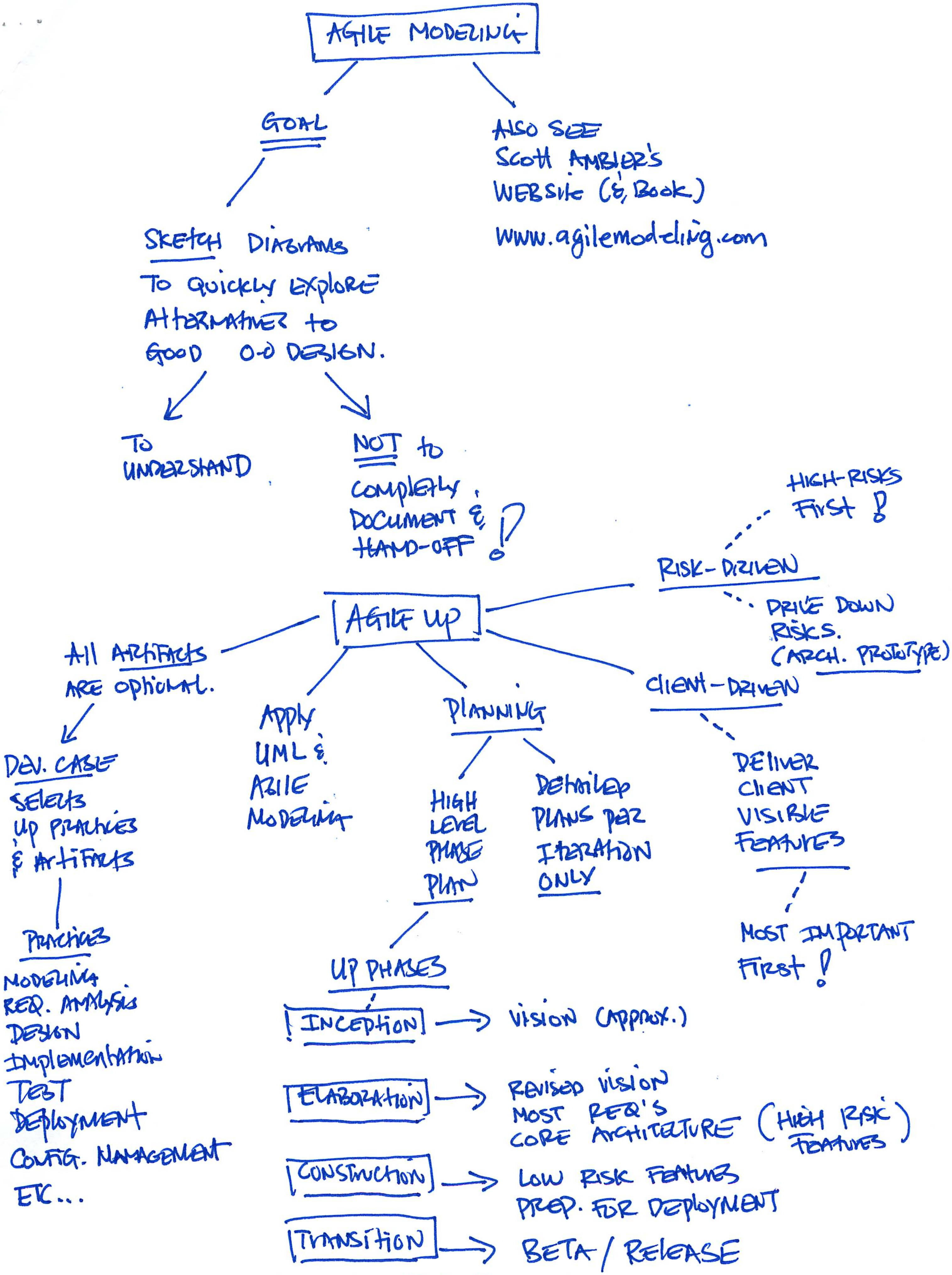
SCRUM
- COMMON WAR ROOM
- DAILY SCRUM

OTHERS
- ITERATIVE
- RISK-DRIVEN

I.E.

LEAN DEV
DSDM
FEATURE-DRIVEN
ADAPTIVE DEV.
ETC...


ITERATIVE DEVELOPMENT
ITERATION
OUTPUT
TESTED &
INTEGRATED
PARTIAL SYSTEM
SHORT "FIXED-LENGHT"
MINI - PROJECTS
(I.E. 3 WEEKS)
INCLUDES
OWN SET OF:
- REQ. ANALYSIS
- DESIGN
- Impl..
- TESTING
- ETC...
ITERATIVE LIFE-CYCLE
3 WEEKS
I1
output
S1
FEEDBACK
& ADAPTATION
I2
output
S2
ETC...
SYSTEM GROWS
INCREMENTALLY
UP
(A.K.A)
ITERATIVE &
EVOLUTIONARY
ITERATIVE &
INCREMENTAL
INFLUENZED
BY EARLY
IDEAS FROM:
- SPIRAL
- EVOLUTIONARY



AGILE MODELING

GOAL

SKETCH DIAGRAMS TO QUICKLY EXPLORE ALTERNATIVES to GOOD O-O DESIGN.

TO UNDERSTAND

NOT to COMPLETELY DOCUMENT & HAND-OFF !

ALSO SEE Scott Ambler's WEBSITE (& Book)
www.agilemodeling.com

AGILE UP

RISK-DRIVEN

HIGH-RISKS FIRST !

DRIVE DOWN RISKS. (ARCH. PROTOTYPE)

CLIENT-DRIVEN

DELIVER CLIENT VISIBLE FEATURES

MOST IMPORTANT FIRST !

All ARTIFACTS ARE optional.

DEV. CASE
SELECTS UP PRACTICES & ARTIFACTS

PRACTICES
MODELING
REQ. ANALYSIS
DESIGN
IMPLEMENTATION
TEST
DEPLOYMENT
CONFIG. MANAGEMENT
ETC...

Apply UML & AGILE MODELING

PLANNING

HIGH LEVEL PHASE PLAN

DETAILED PLANS PER ITERATION ONLY

UP PHASES

INCEPTION → VISION (APPROX.)

ELABORATION → REVISED VISION
MOST REQ'S
CORE ARCHITECTURE (HIGH RISK FEATURES)

CONSTRUCTION → LOW RISK FEATURES
PREP. FOR DEPLOYMENT

TRANSITION → BETA / RELEASE