

Software Engineering
Sapienza Università di Roma

11 September 2020 – Duration 80 mins

- **Be concise and right to the point.** What you write should be understandable by a colleague of yours who just enrolled the MSc in Engineering in CS.

Question 1 (Containerization and micro-services) [15 mins]

- Present and discuss the concept of *containerization*, the differences/similarities with the concepts of virtualization and cloud computing, and the mostly adopted technologies
- Discuss the implications and relationships with the concept of *microservice*

Question 2 (Software architecture and SCRUM) [45 mins]

Assume you are the leading software engineer at *MySoft Inc.* and you are in charge of the analysis, design and realization of an application as **Immuni**. Immuni is the application developed by the Italian Government for contact tracing during COVID, and consists of the mobile app (for contact identification and user interaction) and a back-end where all data are collected, analysis are conducted, notification sent, etc. There is also a Web interface for healthcare operators in which they can insert the results of positive cases, etc.

- a • Design the architecture of the system as you envision it
- b • Assuming a total development time of 3 months, organize the software development life cycle in SCRUM

It is assumed you know Immuni, in case of doubts make your own assumptions, describe them and develop the exercise on the basis of such assumptions

Question 3 (Function Points and COCOMOII) [20 mins]

- Describe the method of Function Points for evaluating software development complexity. Provide all the basic notions and exemplify through examples, whenever possible and appropriate.
- Describe the method of cost and effort estimation commonly referred to as COCOMOII, by also discussing the relationships with Function Points

Q1

a) CONTAINERIZATION IS A TECHNOLOGY THAT ALLOWS TO PACKAGE AN APPLICATION AND ALL ITS DEPENDENCIES INTO A STANDARD UNIT CALLED A CONTAINER. THE CONTAINERS ARE LIGHT, PORTABLE AND ISOLATED FROM EACH OTHER.

EACH CONTAINER HAS ITS OWN SPACE FOR APPLICATIONS AND RESOURCES, SEPARATE FROM THE OTHERS.

VIRTUALIZATION:

- IT USES VIRTUAL MACHINES (VMS), EACH WITH ITS OWN OS.
- MORE RESOURCE-HEAVY THAN CONTAINERS.
- CONTAINERS SHARE THE HOST OS, WHILE VMS DO NOT.

BOTH ENABLE APPLICATION ISOLATION. THEY PROMOTE SCALABILITY AND RAPID PROVISIONING.

CLOUD COMPUTING:

CONTAINERIZATION IS OFTEN USED IN THE CLOUD TO DEPLOY AND MANAGE SCALABLE APPLICATIONS.

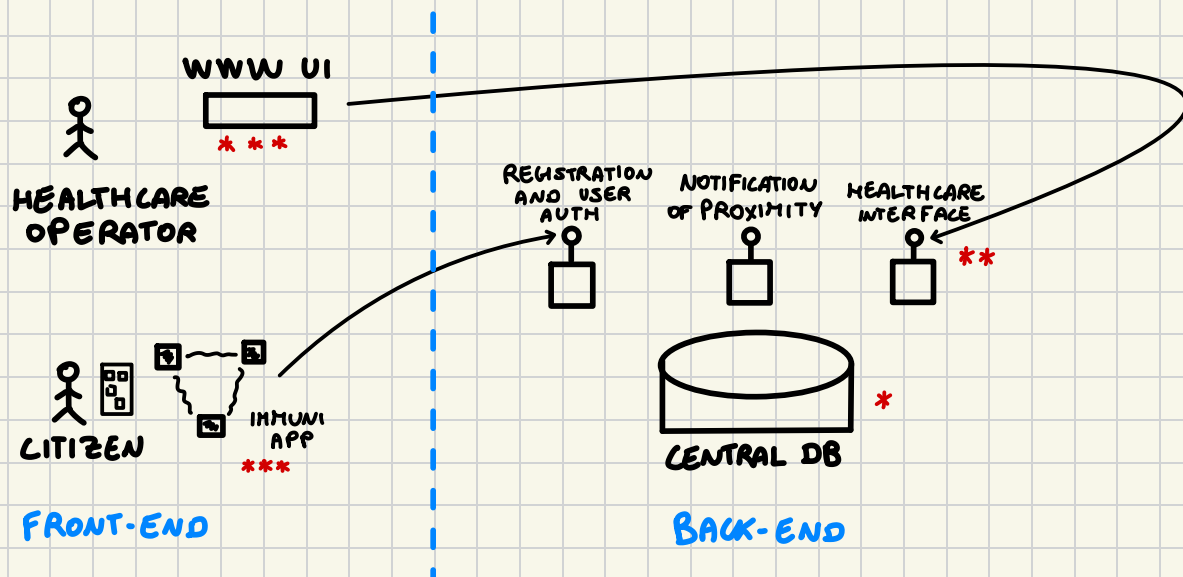
THE CLOUD PROVIDES THE UNDERLYING INFRASTRUCTURE TO RUN CONTAINERS OR VMS.

b) MICROSERVICES ARE A SW ARCHITECTURE IN WHICH APPLICATIONS ARE DIVIDED INTO SMALL INDEPENDENT SERVICES, EACH FOCUSED ON A SPECIFIC FUNCTIONALITY.

CONTAINERS ARE THE IDEAL METHOD FOR IMPLEMENTING MICROSERVICES, THANKS TO THEIR ISOLATION AND PORTABILITY. THEY ALLOW TO SCALE SPECIFIC MICROSERVICES BASED ON LOAD.

Q2

a)



* DATABASE/MODEL HIGH-LEVEL DESIGN (SKETCH OF THE DOMAIN)

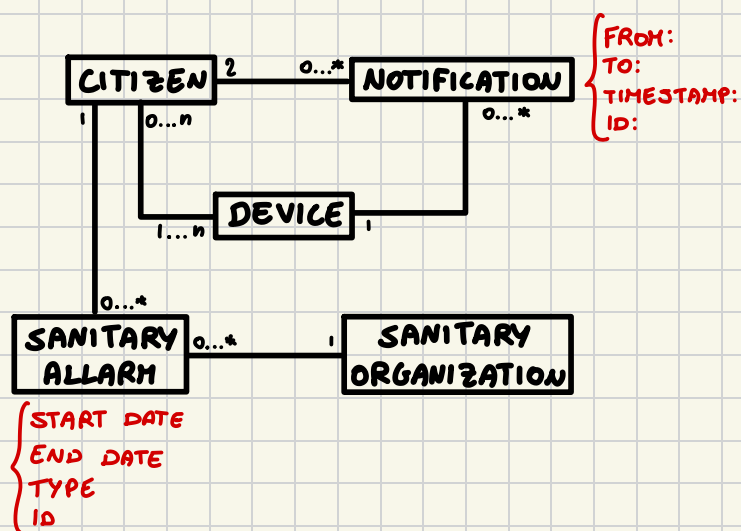


DIAGRAM WRITTEN AS
UML CLASS DIAGRAM

** INTERFACES

REST

/user < GET: ALL USERS
POST: ADD NEW USER

/user/{id} < GET: DETAIL OF THE USER id
PUT: UPDATE DETAIL OF THE USER id

/sanitary allarm < GET: ALL EVENTS
POST: ADD NEW EVENT

/user/{id}/events — GET: ALL EVENTS PERTAINING citizen id

MON (JAKARTA/AQHP)

SPECIFYING AN INTERFACE FOR ASYNC MEANS TO DEFINE:

- THE QUEUES/TOPICS TO BE USED
- THE STRUCTURE OF THE MESSAGES

1. PROXIMITY-QUEUE

CLIENTS (MOBILE DEVICES)
PUSH NOTIFICATIONS OF PROXIMITY

EVENT ID
DEVICE FROM
DEVICE TO
TIMESTAMP
LOCATION

STRUCTURE OF
THE MESSAGES

2. ALERT QUEUE

BACK-END NOTIFY A CITIZEN OF
A POSITIVITY OF ONE OF YOUR CONTACTS

DEVICE ID
CITIZEN ID
DATE FROM
DATE TO
TYPE

THERE IS ADDITIONALLY A BLUETOOTH-BASED COMMUNICATION FOR PROXIMITY BASED ON SOCKET.

b) SCRUM 3 MONTHS

SPRINT OF 2 WEEKS → UP TO 6 SPRINTS

1) AS A CITIZEN

I'D LIKE TO INSTALL AND
REGISTER TO IMMUNI SYSTEM.

FIRST VERSION OF THE APP.

MICROSERVICE FOR REGISTRATION
AND OUTH.

PORTION OF THE MODEL FOR CITIZEN
AND DEVICE.

2) AS A CITIZEN

I'D LIKE TO NOTIFY
PROXIMITY THROUGH
MY DEVICE.

SECOND VERSION OF THE APP USING BT
FOR PROXIMITY.

PART OF THE MICROSERVICE FOR NOTIFICATION
PART OF THE DATABASE MODEL ABOUT
PROXIMITY NOTIFICATION.

3) AS AN HEALTHCARE OPERATOR
I WANT TO INSERT AN
ALLARM

WWW INTERFACE FOR HEALTHCARE

PART OF THE DB ABOUT SANITARY ALLARM.

MICROSERVICE FOR HEALTHCARE.

4) AS AN HEALTHCARE OPERATOR
I WANT TO NOTIFY CITIZENS
OF THEIR RISK

DEVELOP THE ALGO RECONSTRUCTING
POSITIVITY / PROXIMITY.

NOTIFY WWW INTERFACE OF ALLARMS.

5) AS THE GOVERNMENT
I WANT TO NOTIFY CITIZENS
THROUGH THE APP

THIRD VERSION OF THE APP.

COMPLETE THE NOTIFICATION
MICROSERVICE.

3 VERSION OF THE APP

2 VERSION OF THE WWW INTERFACE

- a) FP ARE A MEASUREMENT TECHNIQUE USED TO EVALUATE THE COMPLEXITY AND SIZE OF A SW SYSTEM BASED ON THE FUNCTIONALITY PROVIDED TO USERS.

THEY REPRESENT A MEASUREMENT INDEPENDENT OF THE TECHNOLOGY USED, BASED ON THE VOLUME OF FUNCTIONALITY THAT THE SW SYSTEM MUST PROVIDE.

MAIN COMPONENTS.

EXTERNAL INPUTS (EI): DATA OR INFORMATION ENTERED BY THE USER.

EXTERNAL OUTPUTS (EO): DATA OR INFORMATION RETURNED TO THE USER.

EXTERNAL QUERIES (EQ): DATA REQUEST ANSWERED IMMEDIATELY.

INTERNAL LOGICAL FILES (ILF): DATA STORES MANAGED BY THE APPLICATION.

EXTERNAL INTERFACE FILES (EIF): REFERENCED BUT UNMANAGED DATA STORES.

EACH COMPONENT IS CLASSIFIED AS SIMPLE, MEDIUM OR COMPLEX, CONSULTING THE RELATIVE COST TABLE.

- b) COCOMO IS A COST AND EFFORT ESTIMATION MODEL FOR SW DEVELOPMENT, BASED ON EMPIRICAL AND QUANTITATIVE PARAMETERS SUCH AS SW SIZE AND VARIOUS FACTORS THAT INFLUENCE DEVELOPMENT.

THE EFFORT IS CALCULATED IN PERSON-MONTHS (PM) USING

$$PM = A \times (SIZE)^E \times \prod AF_i$$

$$E = B + 0.01 \times \sum_{i=1}^5 SF_i$$

$$F = D + 0.2 (E - B)$$

IN COCOMO II THERE ARE TWO MAIN MODELS:

- **EARLY DESIGN MODEL:** USEFUL FOR OBTAINING ESTIMATES DURING THE DEVELOPMENT ANALYSIS PHASE FROM SCRATCH. IT'S APPLIED WHEN THE USER REQUIREMENTS HAVE BEEN DEFINED AND THE DESIGN BEGINS.

7 ADJUSTMENT FACTORS ARE USED:

$$PM = A \times (SIZE)^E \times \prod_{i=1}^7 AF_i$$

- **POST-ARCHITECTURE MODEL:** IT'S USED WHEN THE PROJECT IS FINISHED. 17 ADJUSTMENT FACTORS ARE USED:

$$PM = A \times (SIZE)^E \times \prod_{i=1}^{17} AF_i$$

FP FOCUS ON THE FUNCTIONAL COMPLEXITY OF THE SW, AND COCOMO TRASLATES THIS COMPLEXITY INTO ESTIMATED COST, EFFORT AND DURATION, CONSIDERING ADDITIONAL FACTORS.