Chapter 2

Classification of Reqs.

1- Functional Considerations.

Reqs. that fulfil. busines logic and main functionality of software and deliver results to end-users.

2- Product construction considerations.

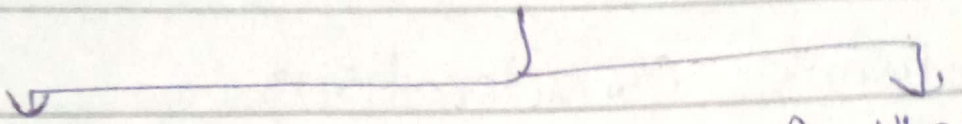
Requirements related to construction of software ie which stack or programming Language to use or which will be deployment platform

3- Source consideration

reqs are provided from diff. sources, this classification is based on agencies that provide reqs. or the person who provide it.

application domain
stakeholder
document
Existing sys.

Classification of Reqs. based on Functionality Considerations.



Core func. reqs

(75%)

↓

without these product is not useful to user These must be fulfilled. The main purpose is to fulfil these requirements.

Ancillary func. reqs. (non-func.)

(25%)

↓

without these product will work but may be not so good. The team take care of these reqs.

Ancillary Func. Reqs.

Statutory Func. → Related to standards, which standard is being followed i.e; IEEE, SPINs etc. This make sure that person with disabilities can also be able to use product.

Safety Func. → Func related to make the software safe to use i.e in medical product patient must stay safe from product.

Security Func. while safety func. is to protect user from product, security is safety from external attacks. Protections from hackers, malicious use of insiders and data theft via Internet.

Usability func. It was referred to as User-friendliness Reffered to GUI.

Data Integrity data provided by the software should be concise and correct.

Response time How much time is take to perform a task.

Footprint constraints harddisk space req. to install .exe or .apk

Memory Constraints Memory /Ram /GPU req. during execution of software product.

Fault Tolerant should be fault tolerant, an error should display a proper msg.

Reliability. program should be reliable if light is gone program should be autosaved.

Feel-good func. This func. is making user interface look sexy. The idea is to make user feel good while using software. The GUI should be sexy.

Esteem func. This func. brings pride to users. For eg:- a Rolex watch delivers same core functionality as other watch but is more sexy. In software adding functionality that no other have. (unique)

Inverse req. user restrictions that should not be in product.

On-ownership Func.

Compative edge example bidding in carcem. Software should be good from compatitors. (Compatative Edge).

→ Comparison

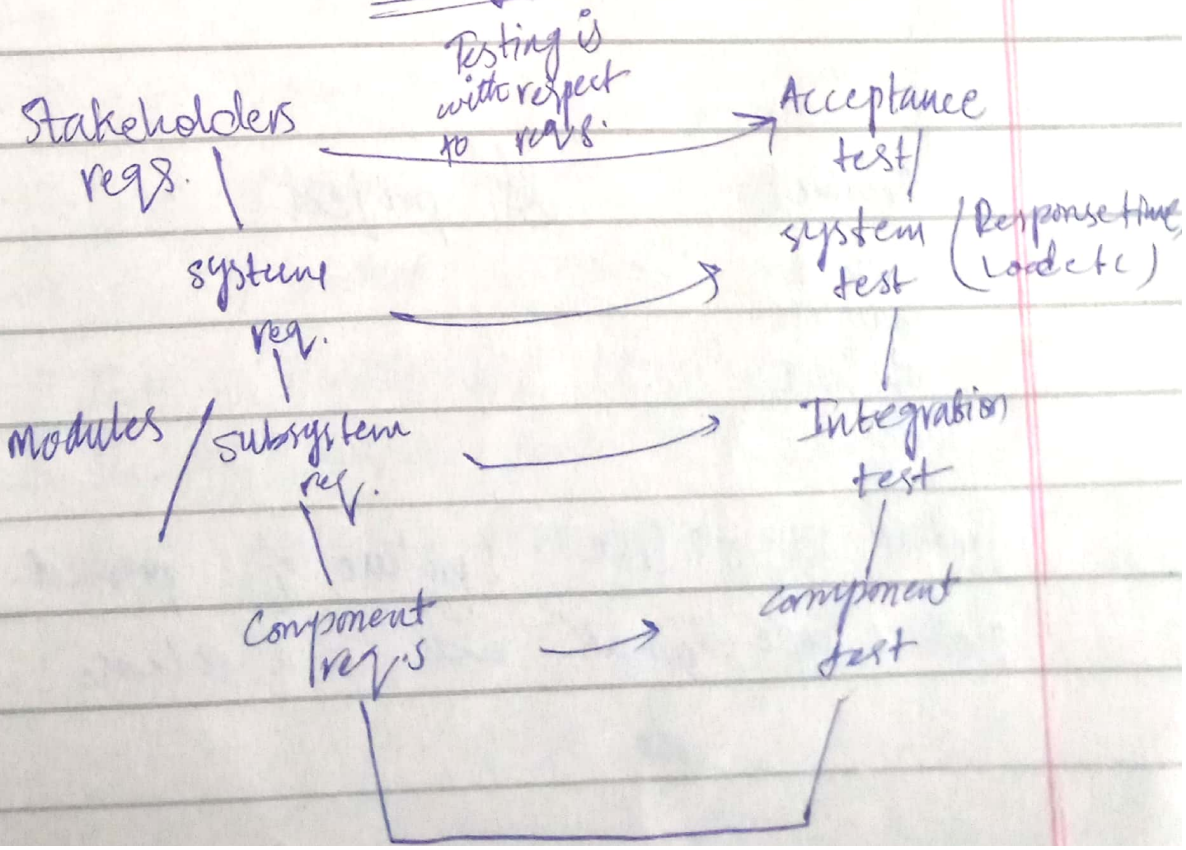
Req. process

Quality of process (If reqs are clear)
↓
Quality of product (good product)

1) Reqs Elicitation
2) Reqs Negotiation / Analysis
3) Reqs specification / documentation (.doc) (Artifact / document)
4) Req. validation

CMMI → levels

Lifecycle



Business level Reqs.

- business need
- criteria of success
- a project business need, as well as the criteria of its success.
- may be financial purpose/ increase sale by 15%
- might come from funding sponsors, corporate executives, marketing manager or product visionos

Product & project

↓

complete software

Product vision → true picture of product.

product scope → what will a release do.

Buss. Req.s

1) Background. (current system info./drawbacks)

2) Buss. opportunity. (what is buss. opp. for example user increase)
   ⟹ (comparison of new with prev.

3) Buss. objective.
   It should be quantitative and measurable ways.

4) Vision statement
   It should contain.

Vision sample

→ For (target (audience) (For a small company)
→ who (need) (to ensure they meet employment legislation Boards)
→ The (product name) (Staff squared)
→ is (product category) (eg: HR)
→ That (features/key benefits) (provided to customer)
→ Unlike (compare competitors) / existing
→ Our product (advantages) / uniqueness / prominent features

→ Success metrices → criteria on which product is either good or not.

indicators that ensure that product is successfully done.

↓

in vision statement

→ Business Risk => always related to competitators.

→ Business assumptions and dependencies.

→ from $15%. to $25%.

→ how much visitors should be ?

Assumption ↳ → how much should be daily monthly sale.

should be correct

all related to business

Scope Representation

→ Context diagram

↳ Notation

Entire system in Circle ○

External entities in ▭ (always noun)

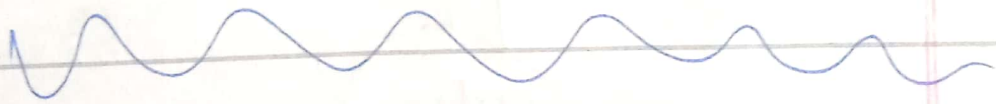
↳ user classes

↳ Organization

↳ Other system

↳ Hardware

Flow of data represented by →



Ecosystem map (Tool)

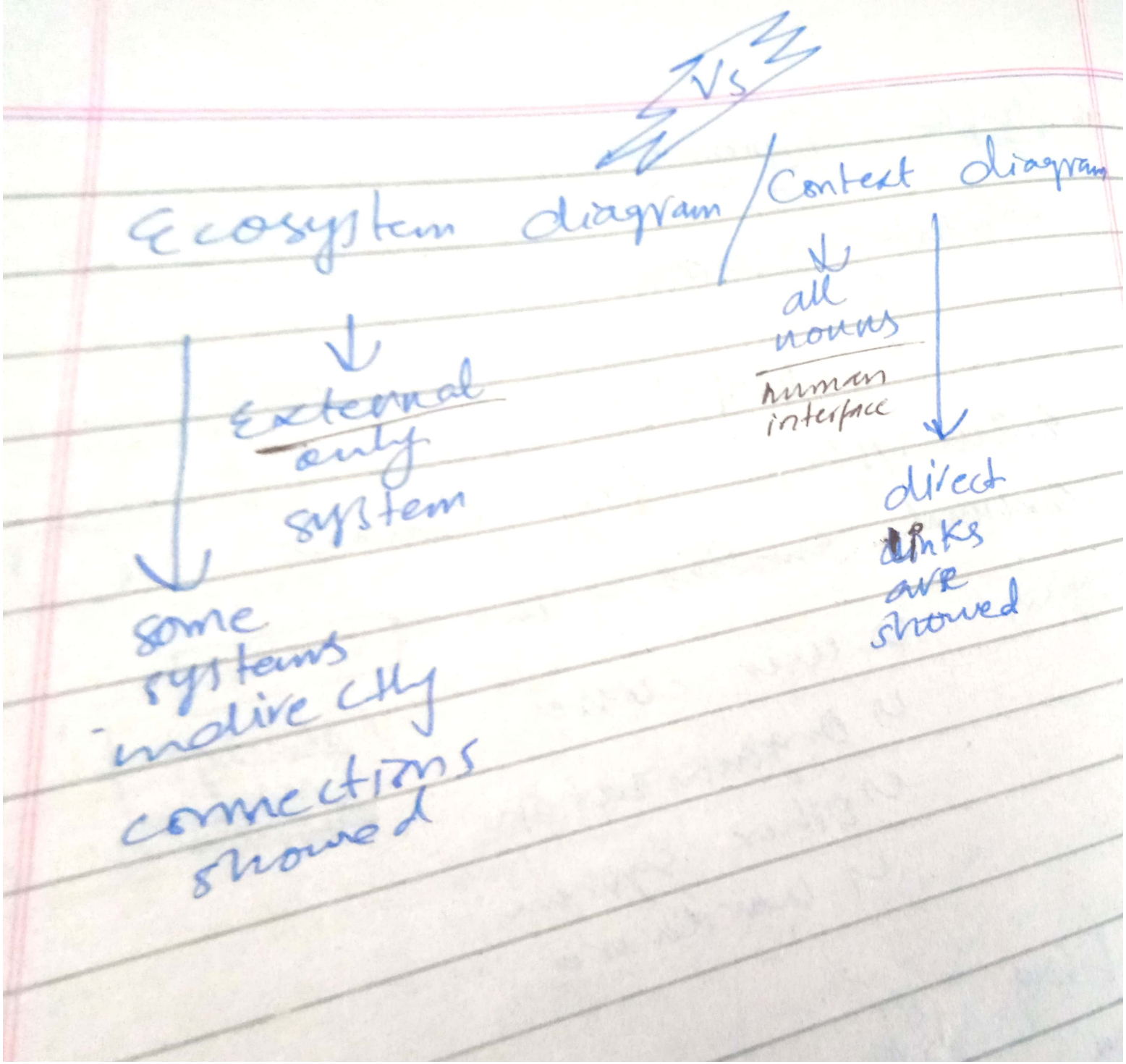
we know interaction of our system with other systems

(system interaction)

Bold Box → Entire system

Box → External system

→ → Flow of data



Scope Representation :-

→ Context diagram

→ Ecosystem map

→ Feature Tree

all features are listed in form of tree. all branches represent Feature

→ level-I → main Feature eg search

→ level-II
→ level-III
→ sub Features → search by Id

~~sys~~ (MC chemical)

Blue box L-I

Grey box L-II

Simple L-III

Event List / Triggers :-

event list identifies external events that could trigger behaviour in system.

Triggers

- By user → Business trigger
- Time Trigger (Automatic BUY gen.)
- Signal events (credit card)

context / ecosystem

- External Actors
- system involved

Ch 6
Bus. req

Req. Elicitation

- Personal interview
- Questionairy
- survey
- observation
- Demonstration of product — Prototypes of product itself.
- Brainstorming

Elicitation → Draw out → customer

Gathering → come together/assemble from sources → doc.s / existing software

Stakeholder	Technique
if (same er < 1 (1-3))	Interview
else if (>100) students (>100)	other Techniques Survey / Questionnaires

number designation schedul