SOFTWARE RE-ENGINEERING

Date: 17th Oct, 2022

R	E
01- Requirements	01- Evolution
02- Roles	02- Emerging
03- Rules and Regulations	03- Efficiency
04- Recruitment	04- Effectiveness
05- Reverse Engineering	05- Errors
06- Refactoring	06- Encryption
07- Reusability	07- Ease of use
08- Recycling	08- E-commerce
09- Redundancy	09- Enhancement

→ Re-engineering increases performance, maturity, structure, architecture

→ Facebook, Samsung, iOS (15.6 → 16.0) ⇒ less ease of use.

→ A system can be efficient but can have bad performance.

Difference between Reverse Engineering & Re-engineering

REVERSE ENGINEERING	RE-ENGINEERING
→ A finished product is deconstructed and tested to discover how it works.	→ Involves the investigation and redesign of individual components
→ Info about design & composition is extracted.	→ Improve particular area of performance
→ Retracing of design steps	→ Improve functionality
→ Inspect product for quality issues.	→ Reduce operation cost/inc efficiency
→ competing suppliers perform	→ Add new features.
	→ Minimise downtime
	→ process of s/w development

Date: 19th Oct, 2022

reverse engineering to learn how it was built & how it works.

→ To increase interoperability

→ To learn what went wrong

→ Improvement with new tech.

→ Through deductive reasoning understand how a product was made.

→ Rediscovering the design, req spec & func of a product from an analysis of its code.

→ Improving understandability of system

done to improve the maintainability of software.

→ Examination and alteration of a system to reconstitute its new form.

→ Includes sub-processes like reverse engg., forward engg., reconstructing etc.

→ reorganizing & modifying existing s/w.

→ To dev a cost-effective option for sys evolution.

→ Reduced risk & cost, revelation of business rules

e.g. market trend changing

ADVANTAGES:-

inc processing speed, size of chip remain same

01- Moore's law (no of transistors on a chip double every year)

02- Software vs Hardware (e.g. see system Windows XP)

03- Successful projects ratio

04- Companies competitive environment.

Nokia couldn't reengineer themselves. Didn't follow market trend

DISADVANTAGES:-

01- Cost (time, space, monetary)

02- Performance (accuracy) may not be enhanced.

03- Customer satisfaction.

04- Market trends ~~and~~ pressure and changes.

Date: 23rd Oct, 2022

STRENGTHS OF NOKIA:-

01- Experience Base (151 years old)
02- Experienced and Technologically advanced people.
03- Large and strong distribution channel.
04- Strong reputation on durability. Monopoly.
05- User-friendly accessories.
06- Resell value remains high.
07- Operating in many countries.

WEAKNESSES OF NOKIA:

01- Slow reaction to competition.
02- Poorly-designed smartphones
03- Low performance of smartphones.
04- Poor after-sale service.
05- Low sensitivity to industrial changes.

Why it fell behind in the market? Symbian

→ Nokia failed to realize the need to adopt Android Operating system for its mobile devices as Android is more attractive.
→ Failed to devise a strategy to compete against competitors like Apple, Samsung, Oppo etc.
→ They overestimated their brand value.
→ They were under the impression that people

wouldn't accept touch screen phones and would continue with QWERTY layout!

→ Launched their own OS Symbian but it was nothing infront of Android!

→ Ill-timed deal with Microsoft.

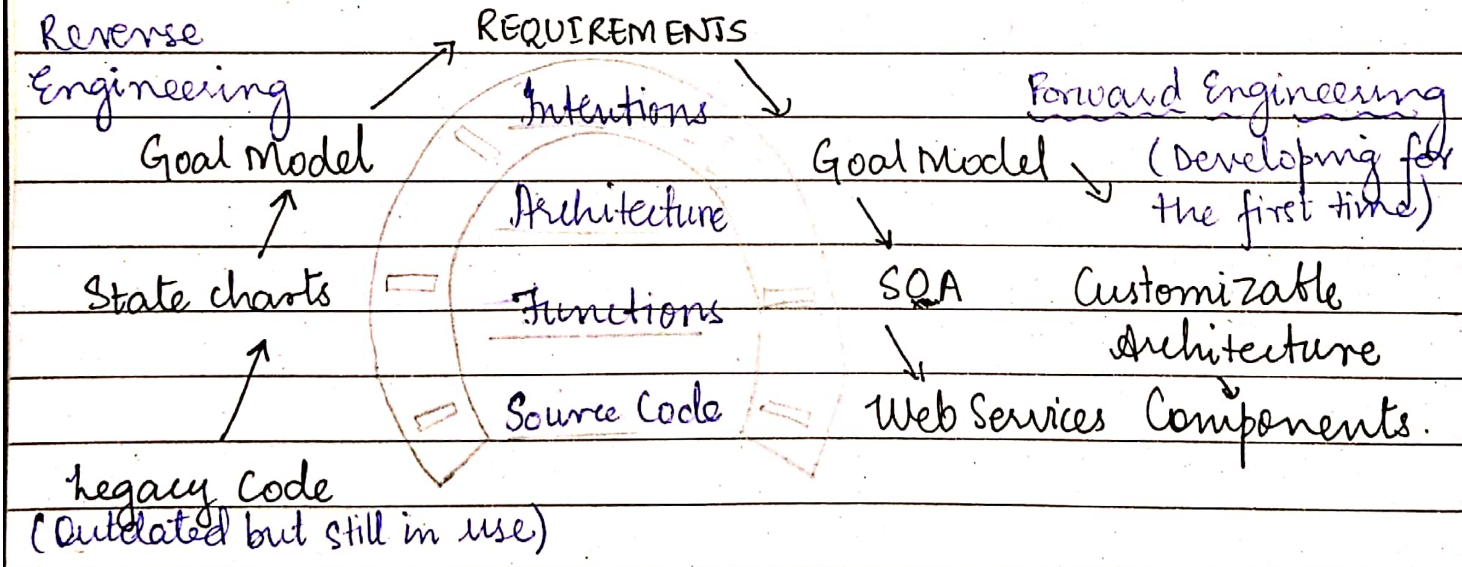
→ Unsuccessful marketing strategy of umbrella branding. (Samsung S Series)

→ Famous for hardware and didn't pay much attention to software.

→ Lack of innovation. No 5G in era of 4G.

Intentions, Functions, SC, Archi

SOFTWARE RE-ENGINEERING HORSE SHOE :-



→ Horse is powerful, and horseshoe depicts the power of SR.

→ Re-engineering starts from requirements.

→ If legacy code is discarded, the product will need to made from scratch.

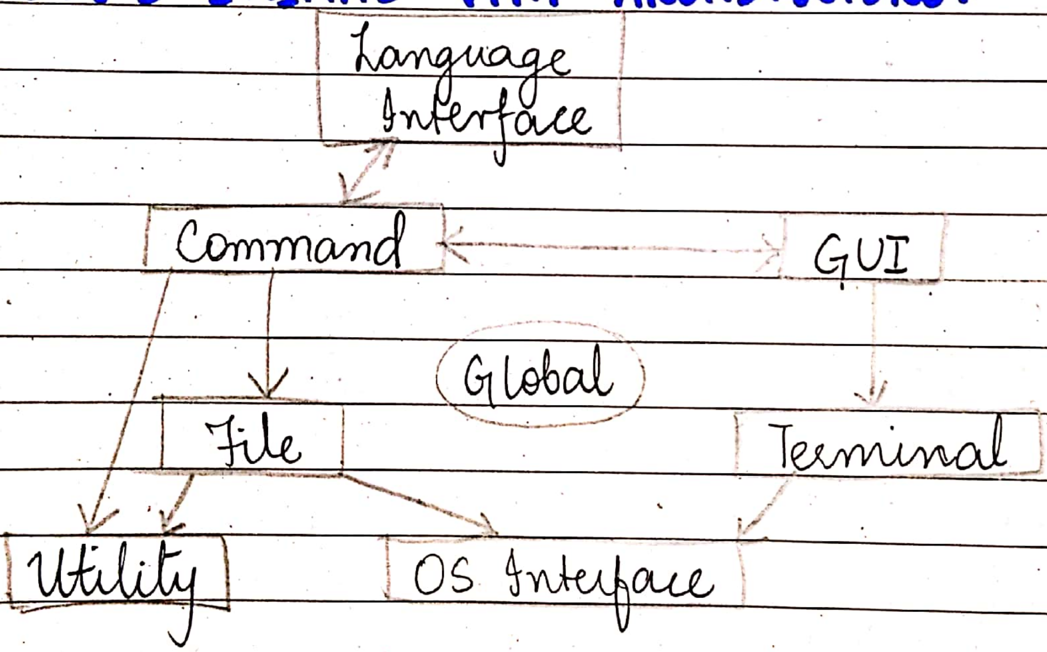
Date: 31st Oct, 2022

→ Requirements are outside b/c they are gateway.
→ Physical, Mental, Psychological, Sleep Deprivation
Medical Issues, Eating habits

ViM (Architecture)
↓ (Vi Improved)

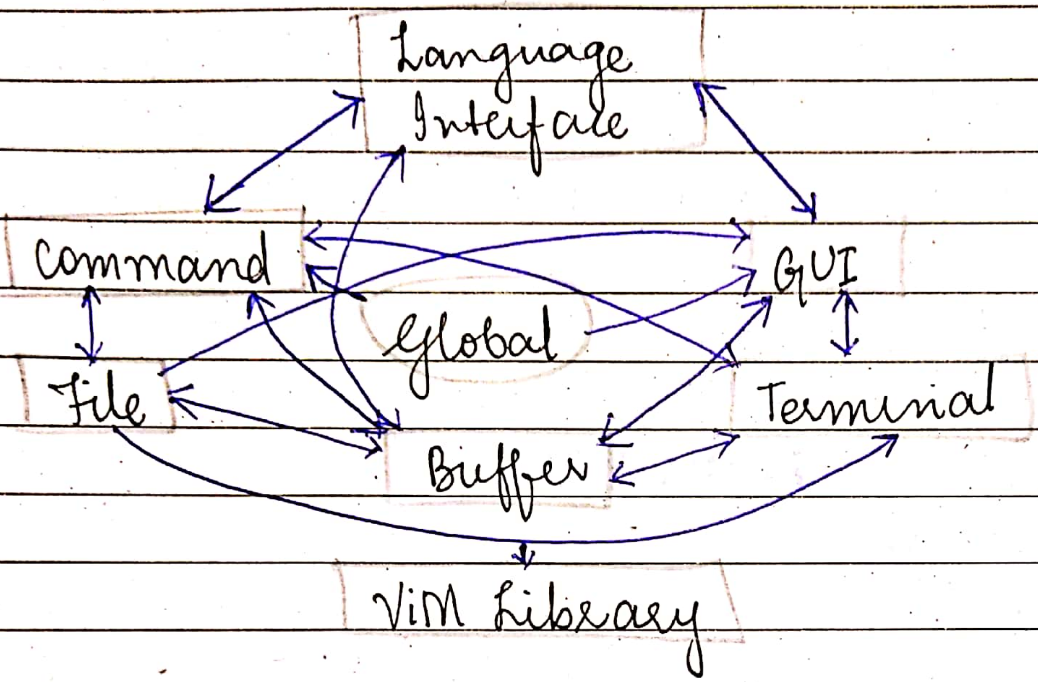
→ Developed in C

LEE'S INITIAL ViM ARCHITECTURE:-



→ Security is more as its has only one entrance end.
→ Simplicity. Less complex.
→ Communication gap (One-way communication).
→ Minimal redundancy.
→ More time complexity in terms of cost.
→ Less space complexity in terms of cost.
→ Portable, Scalable, Adoptable
→ Single point of failure.
→ Global isolation (no inside comm, no contact with outside)

AFTER ARCHITECTURAL REPAIR OF OPEN-SOURCE S/W.



→ Every module can able to communicate with other module.

→ Communication gap bridged.

→ No single point of failure Disadv

→ Less security due to more communication.

→ No global isolation (whole sys connected to library)

→ More probability of errors. Disadv More LOCs.

→ Costly in terms of space. Disadv

→ Less time b/c of direct link.

→ Less Scalable. → Authenticity threat

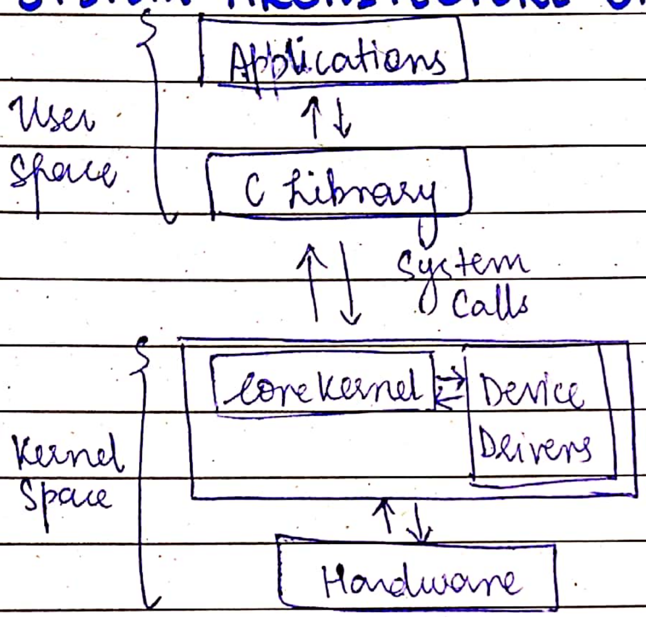
→ ~~Less~~ More time to be maintainable.

→ Performance threats

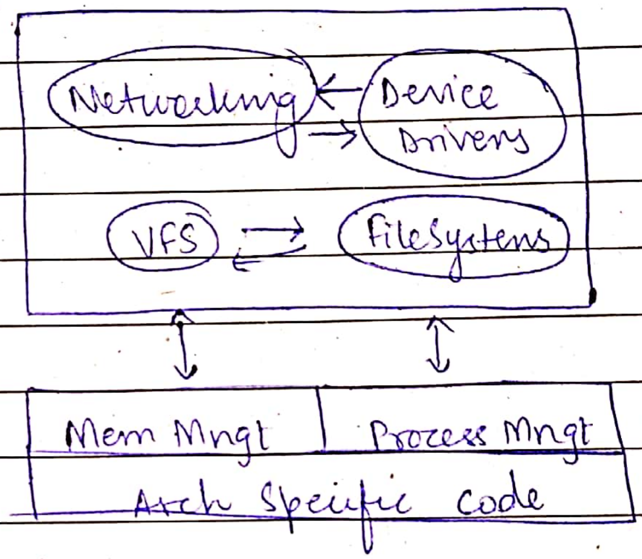
Q1= Compare two architectures.
what are before and after.

Q2= Legacy projects? Are they still in use
If yes then why?

SYSTEM ARCHITECTURE OF KALI LINUX:-



Kernel in depth =>



-> User space => related to user config part (App & related libraries)

-> Kernel Level => main core of Linux.

INSTAGRAM ALGORITHM CHANGES:

01) Introduced new option of Favorites Home feed view: a new chronological feed for more customized experiences.

02) More suggested content.

03) Prioritization of original content.

04) De-prioritization of content with visible watermarks of apps.

05) Improved content categorization

06) Introduction of short form content (Reels)

07) Instagram Music with synchronized lyrics.

08) Auto-subtitles on stories.

0 Instagram story stickers.

Date:

UBER..

01) Trip Riders & Drivers can see list of other trips happening nearby. Ability to pick another trip that works well with them.

02) Splitting the fare.

03) Multiple drop-off & pick-up locations.

04) Speaking with safety agents (like 911 texts)

05) Sync meetings with calender shortcuts.

06) In-app translations.

Antivirus is itself a virus.