

Ethical issues in scientific research

- Conflicts of interest (possible effect on research outcomes by researchers' interest)
- Data manipulation (creating research more interesting or bias in the findings)
- Privacy violations (disclosing or misuse personal data without consent)
- Misrepresentation of research findings (incomplete or misleading manner)

Wrong engineering practices

1. Ignoring safety standards
2. Poor material selection
3. Falsifying test results
4. Neglecting environmental impact
5. Ignoring maintenance protocols
6. Designing for cost over quality
7. Using outdated or inadequate technology
8. Disregarding ethical considerations
9. Lack of transparency
10. Lack of proper documentation
11. Inadequate risk assessment

Ethical principles in scientific research

- Autonomy
- Informed consent
- Honesty
- Confidentiality and privacy
- Justice
- Nonmaleficence (Do no harm)
- Beneficence

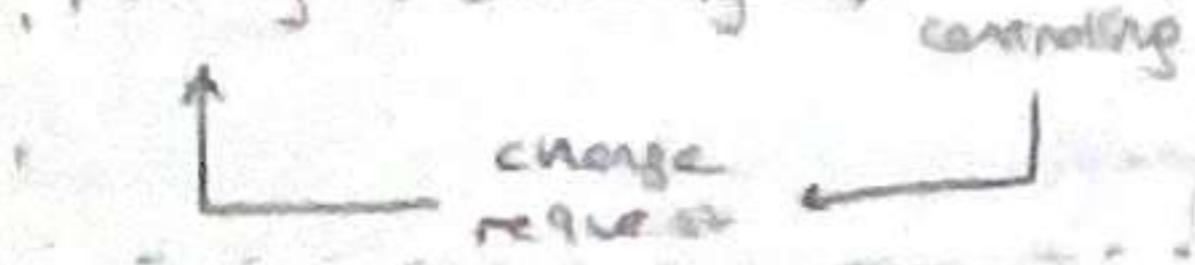
13th presentation

- IT infrastructure invisible foundation makes technology and digital services work, ensures systems are: stable, secure and always available
→ 5G, AI, and cloud computing
- servers run applications
 - storage keep data safe
 - database organize and manage information
 - networks connects everything together

Project management

Planning, organizing, and managing resources to deliver a unique product, service, or result within defined constraints
quality → scope, cost, time

Initiating → planning → executing → monitoring controlling → closing



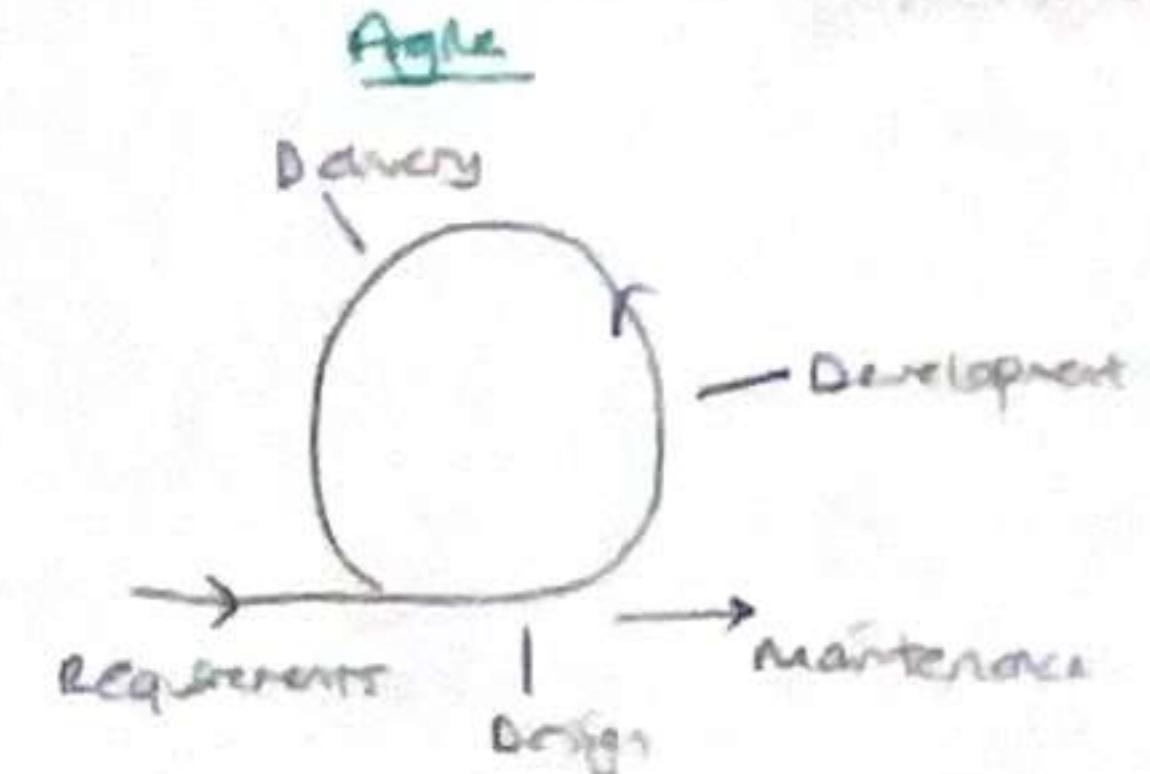
Frameworks and methodologies

eg Agile, waterfall, lean, scrum, critical path method (CPM)

Waterfall

- Requirements
+
Design
+
Development
+
Delivery
+
Maintenance

(step by step structure)



(flexibility, short cycles, continuous imp)

Agile

- + Stakeholder M.
• understand needs
• engage regularly
• clear, jargon-free lay
+ tribe M. + communication M.
• potential risks
• analyze impact
• develop response
• set frequency
• present issues
plan

Common tools: jira (agile boards), MS project (scheduling), trello (task tracking), confluence / sharepoint (collab, documentation), excel / Power BI (dashboards, KPIs)

Business Process Management (BPM)

Approach to designing, executing, monitoring and improving business processes to achieve organizational goals

- increase efficiency
- reduce manual efforts / errors
- processes are modeled and monitored
- 1) Design 2) Model 3) Execute 4) Monitor 5) Optimize
(tuning) (simulate) (action) (measure) (improve)

BPM tools and techniques

Visualize, automate, and continuously

Standardization: define a single efficient way to perform each task

Continuous improvement (kaizen): make small ongoing enhancements

Automation: replace manual steps with technology

Measurement: track time, cost, and performance to the bottleneck

Optimization: simplify complex workflows to reduce waste and effort

Project management helps teams to achieve defined goals
Process management (BPM) keeps daily operations efficient and measurable

Waterfall follows a linear and sequential structure

Agile emphasizes flexibility, iterative development and can implement through short cycles

2nd presentation

Software development

- Process of designing, coding, testing, deploying and maintaining software applications to solve real-world problems

Software engineer / developer

- writes, tests and maintains code
- Specializing in frontend, backend or full-stack development

Frontend

- User-facing part of an app (UI, UX, visuals)

Backend

- Server-side logic, databases, and application processing

Software architect

- Designs overall system structure, technology stack, scalability, and security of software systems

QA engineer (Quality Assurance)

- Tests software to detect bugs and ensure quality
- Creates automated and manual test plans

DevOps engineer

- Bridges development and operations teams
- Manages CI/CD pipelines, cloud infrastructure, and deployments

CI/CD (Continuous integration / continuous deployment)

- Automated processes builds, tests, and deploy software to improve reliability and speed

UI/UX designer

- Designs user interfaces and improve user experience
- Collaborate with developers to create intuitive applications

Product manager

- Defines the product vision and prioritizes features
- Balance technical and business needs

Data scientist / AI engineer

- Analyzes large datasets, builds machine learning models
- Applies AI solutions to optimize business processes

Team lead / technical lead

- Guides developers, reviews code, ensures technical excellence
- Bridges engineering team and management

Project manager / Scrum master

- Plans, organizes, and monitors software projects
- Uses Agile or Scrum methodologies to track progress

CTO (chief technology officer)

- Defines company's overall technology strategy
- Aligns engineering goals with business objectives

API (Application programming interface)

- Enables communication between software components

Version control (git)

- Tracking and managing changes in code

3rd presentation

Product management

- Voice of society, product development, marketing, sales, business operations, support, finance, customer success, leadership and strategy

(**Product manager**: A professional responsible for defining what to build, why to build it and how success will be measured, while aligning business, technology, and user needs)

Benchmarking

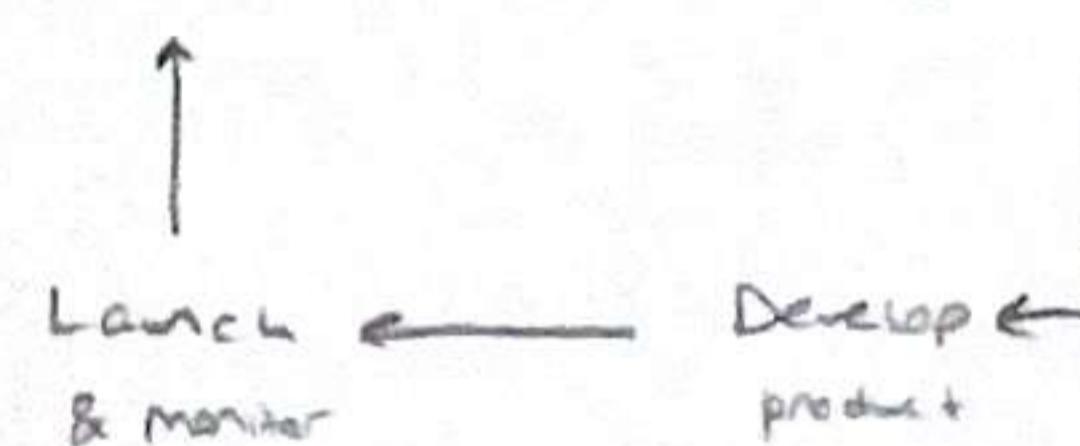
- Comparing products or processes with industry leaders to identify best practices

Product shaping

- Process of turning insights, data, and ideas into a clear, testable product proposal

Product life-cycle development

Define requirements → Design solution



NPS (Net Promoter Score)

- Measures customer loyalty by asking "How likely are you to recommend us to a friend, ranges from -10 to +10?"

Retention

- The percentage of users who continue using your product over time. Show how well you keep them

Churn rate

- The percentage of customers stop using your product within a given time period

Conversion rate

- The ratio of users who take a desired action out of total

RICE framework

- A prioritization method: Reach x Impact x Confidence + Effort helps decide what to build first

MOSCoW method

- Prioritization list: Must have, Should have, Could have, won't have (for now)

Impact & Effort matrix

- A 2x2 grid used to decide priorities: focus on high impact, low effort

OKR (Objective and key results)

- Goal setting framework combining what you want to achieve and how you'll measure success

Agile

- A flexible way of working that delivers products iteratively, learning and adapting quickly

Sprint

- A short time board cycle where the team builds, tests, and delivers specific tasks

Backlog

- A prioritized list of features, bugs, and tasks waiting to be developed

Roadmap

- A visual plan that shows what will be built and when - aligning teams on priorities and timing
- Strategic timeline that connects the product vision to execution

North Star Metric (NSM)

- Single metric that best captures your product's long-term value for customers

MVP (Minimum Viable Product)

- The smallest version of a product that delivers value and helps validate and refine quickly

A/B test

- An experiment comparing two versions (A and B) to see which performs better

4th generation

Agility

- Ability to adapt changes in the environment and objective changes

Strategy

- High level
- Affects long-term goals, the vision and overall structure of organization

Tactical

- Medium-term plans and processes supports strategic objectives

Operational

- Short-term changes directly affect daily op. tasks and operational workflows

Focus

- Early and frequent delivery of business value
- Continuous improvement of the product and processes
- Flexible scope management
- An empowered team and close collaboration with the customer
- Delivering products that fully meet customer needs

5 Fast sensing 5 Fast decision-making

- Close communication, early feed-back, early failure, focus, clear goals, accelerate measurement, transparency, trust and alignment, continuous improvement

5 Fast maneuvering

- Clear goals, small teams, intra-team alignment, intra-team trust, continuity, automation, modular architecture, autonomy, continuous improvement

Continuous motivation

- Corporate belonging, sense of the importance of the work, responsibility and authority, alignment, self-development, self-actualization, environment and support, doing enjoyable work

Organizational perspective change

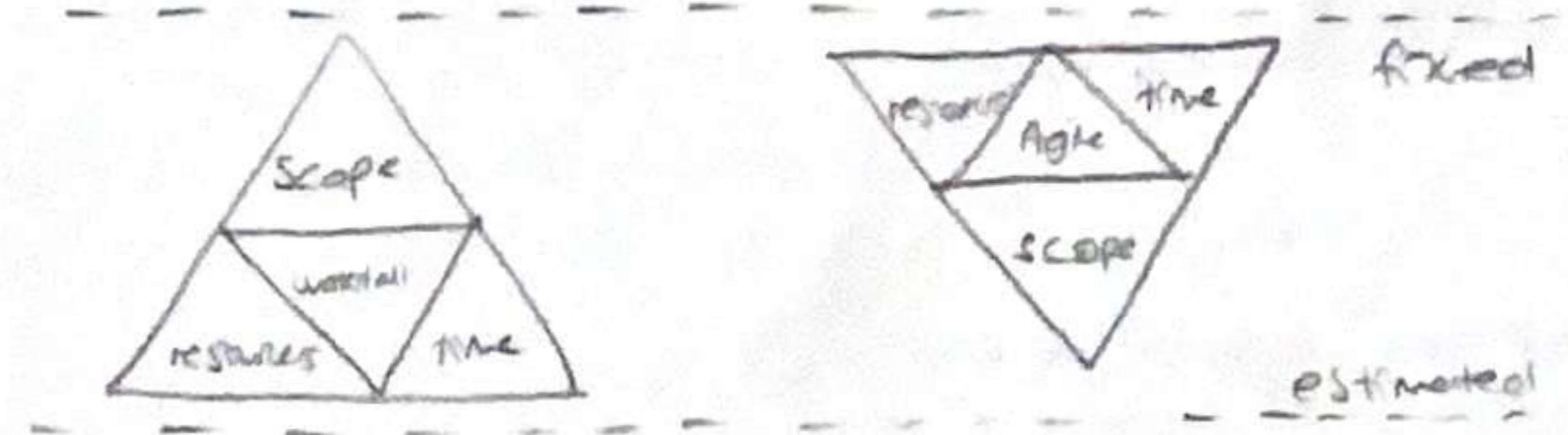
Traditional

- Static
- Function-based
- Plan conformance
- Management
- Control
- parent and control
- rule
- post performance evaluation

Lean/Agile

- dynamic
- Mission-based
- leadership
- control
- empowerment (autonomy)
- mitigate and adapt
- principle
- future performance improvement

Waterfall



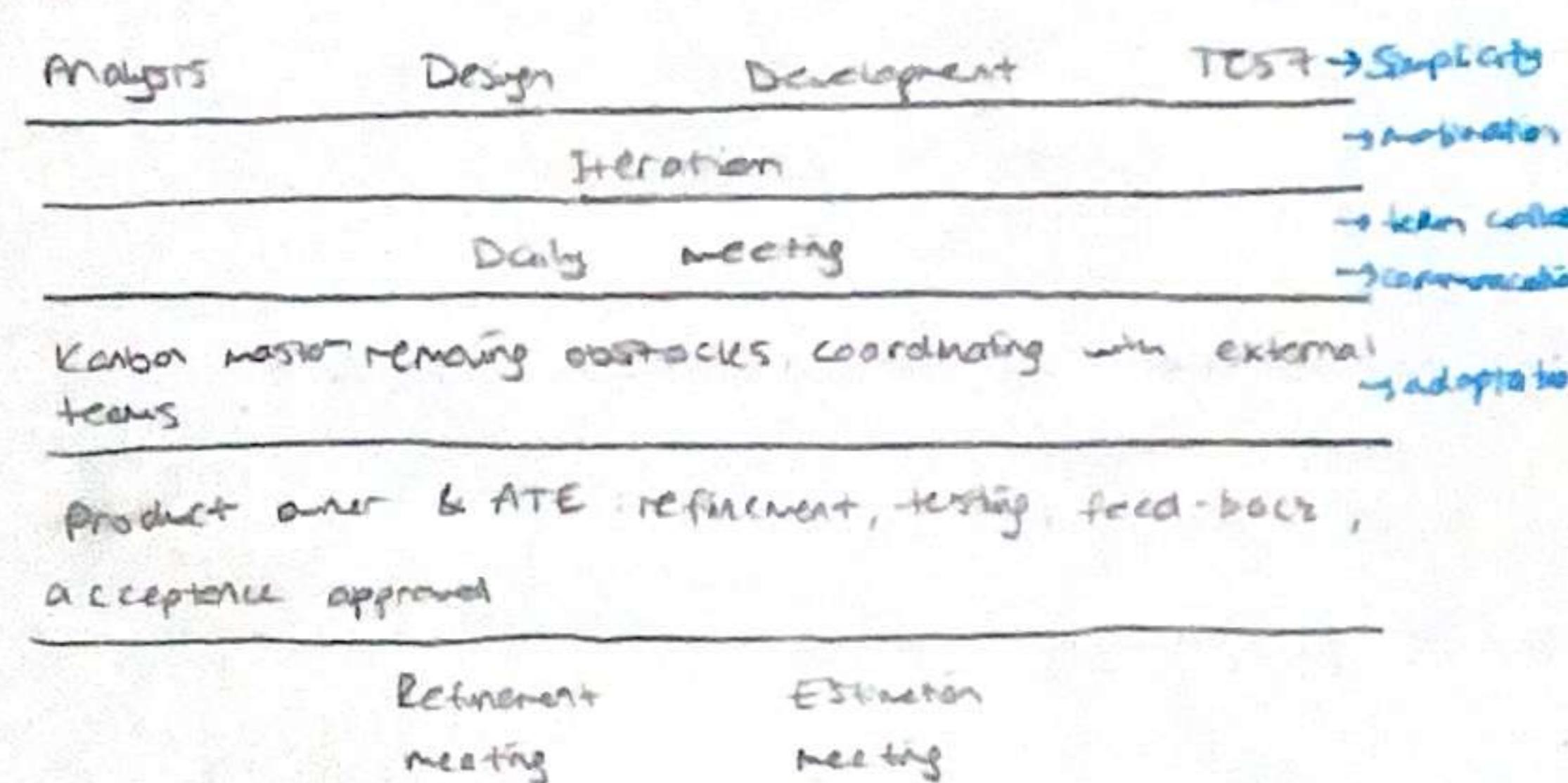
- Ideal for construction and engineering
 - Generally proceeds with prototyping
 - Requires centralized control, as complexity increases, centralized control breaks down
- Requirements → Design → Implementation → Verification → Maintenance

Kanban

- An inventory control system used in just-in-time manufacturing to track production and order raw parts and materials

Kanban team (squad)

- Kanban master, product owner, development team, acceptance testing team



Scrum

- It originated in software development, but it is not exclusive to software

- Iterative planning
- Iterative execution

Core elements (3 pillars)

- Transparency
- Inspection
- Adaptation

Values

- courage
- focus
- commitment
- respect
- openness

Roles

- | | | |
|-----------------|-------------------|------------------------|
| • Product owner | • Product backlog | • Sprint planning |
| • Developers | • Sprint backlog | • Daily scrum |
| • Scrum master | • Increment | • Sprint review |
| | | • Sprint retrospective |
| | | • Sprint |

squad → scrum/kanban master + product owner + developers

Scrum team → scrum master + product owner + developers

Self-managing

Meaning they internally decide who does what, when, and how

Cross-functional

Meaning the members have all the skills necessary to create value each sprint

Product owner

- The person who defines products/services/outcomes, prioritizes by value, and provides approval before completion to reach over and team goals while best fulfilling customer needs
- vision and priorities
 - accessibility
 - knowledgeable
 - authoritative
 - Product management
 - publications and deliveries
 - decisive
 - accountable

ACTIONS

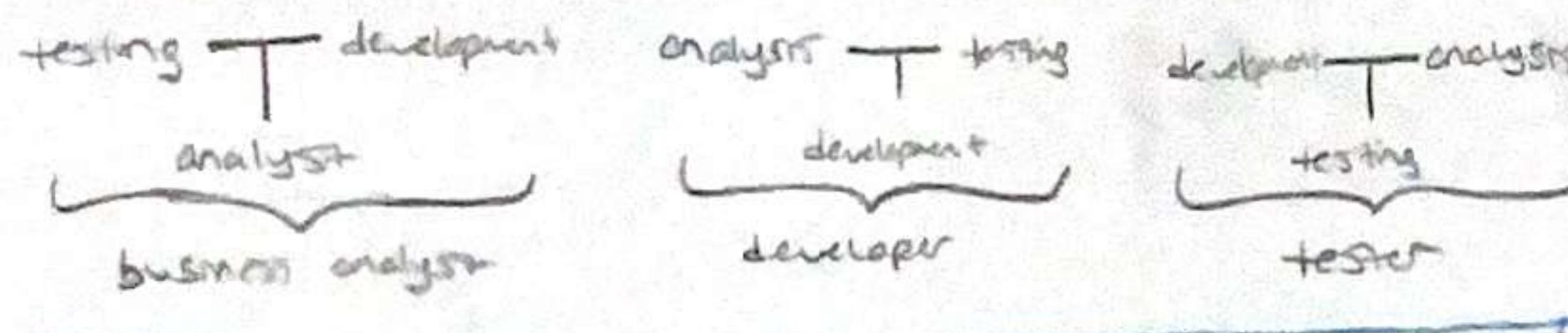
- 1) PO list of features expected in Sprint
- 2) Squad selects items they can complete
- 3) P.O. states the order of priority
- 4) Squad states final commitment what to be done
- 5) Goal is determined by PO and team
- 6) Selected stories are broken down into tasks
- 7) Estimation is performed for tasks
- 8) Team decides how work will be done
- 9) who will do what (later)
- 10) Goal revised if needed by PO and team

Scrum master

- Enriches team agility
 - Facilitates events
 - Demonstrate servant leadership
 - Deliverables and outputs
- ✗ not manager, not impose anything, not full ownership

Scrum team

- Cross-functional work on multiple tasks
- Self-organizing
- how much work to do in a sprint
- Self managing
- determine how they will work
- Collaboration
- work together



Product backlog

- Includes prioritized and sized user stories, technical stories, bugs, and risk response plans

Minimum viable product (MVP)

- Reduces risks
- Prototype for subsequent products

DR: Definition of Ready

- A list of clear conditions that indicate a work item is "Ready" to be taken into team and started

DD: Definition of Done

- A list of objective and measurable criteria that must be met for a work item to be considered truly "Done"

Scrum events

- Sprint (not 4 weeks)
- Sprint planning (1 hour)
- Daily scrum (15 min) → Yesterday? → Today? → Anything blocking?
- Sprint review (2 hours)
- Sprint retrospective (1.5 hours)

Sprint planning

- The sprint goal is defined
- PO, SM, DT
- what to do, how to do
- At the beginning of sprint

Inputs

- Product backlog items
- last delivered product increment
- DT's capacity and past performance

Outputs

- Sprint goal
- Sprint backlog items

ACTIONS

- 1) PO, squad and agile coach discuss whether there were any issues
- 2) Action items for process improvement identified and planned
- 3) Previous action items are checked
- 4) Held after sprint review and before sprint planning
- 5) Opportunity to focus on inspiration and adaptation

Output

- Next sprint backlog items
- Stakeholder suggestions

ACTIONS

- 1) Product demonstrated
- 2) Team focuses on completed work
- 3) Progress is made visible to everyone

Sprint retrospective

- Team evaluates themselves and sprint
- Opportunity to observe themselves and create a plan for improvements in the upcoming sprint
- PO, SM, DT
- Once every sprint

Input

- Previous retrospective
- Action Items

Output

- Improvement suggestions
- Discussion and planning of action items

ACTIONS

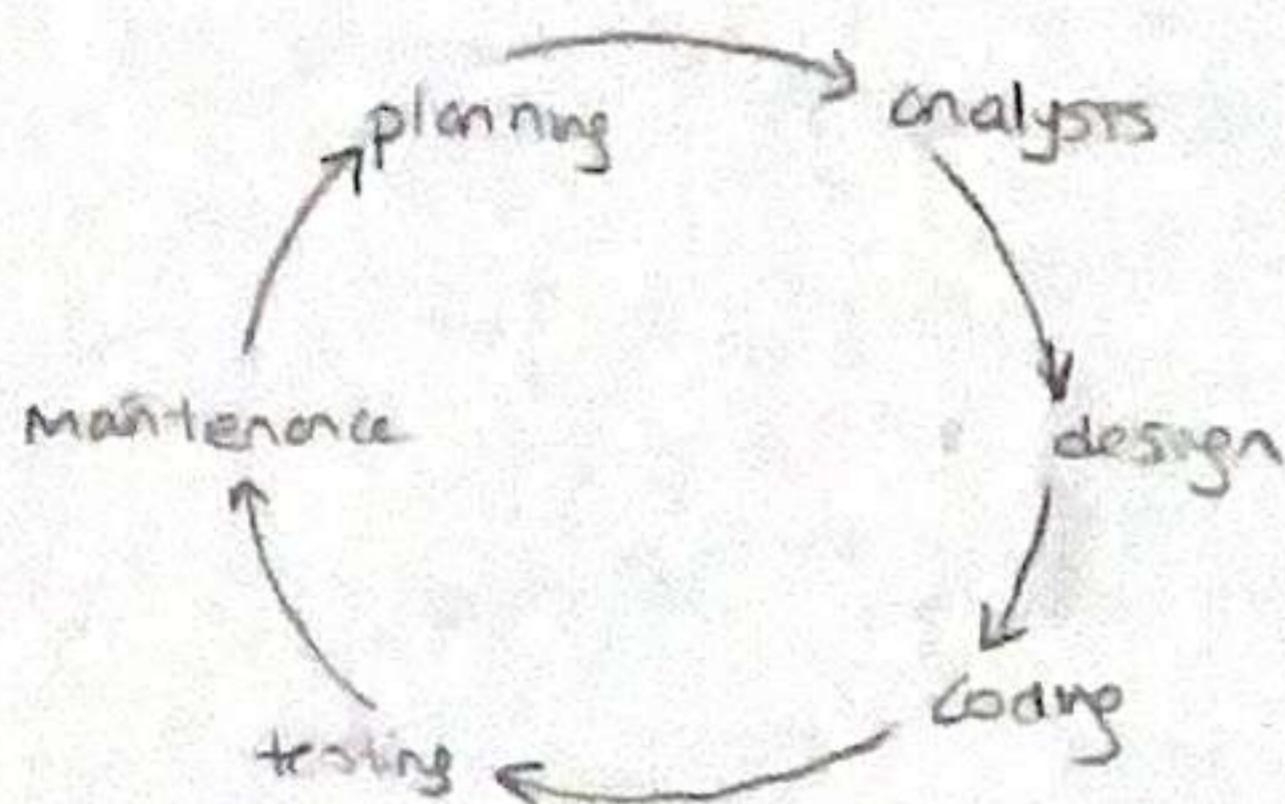
- 1) PO, squad and agile coach discuss whether there were any issues
- 2) Action items for process improvement identified and planned
- 3) Previous action items are checked
- 4) Held after sprint review and before sprint planning
- 5) Opportunity to focus on inspiration and adaptation

5th presentation

Software

- Group of machine commands that enable communication of the electronic devices

Software development life cycle (SDLC)



planning

- Time
- Source
- Requirement

coding

- development
- configuration

analysis

- determining solution
- optimization
- feasibility

design

- system architecture
- user interface

testing

- error detection
- quality standards
- evaluation
- user decisions

maintenance

- continuity
- controlling

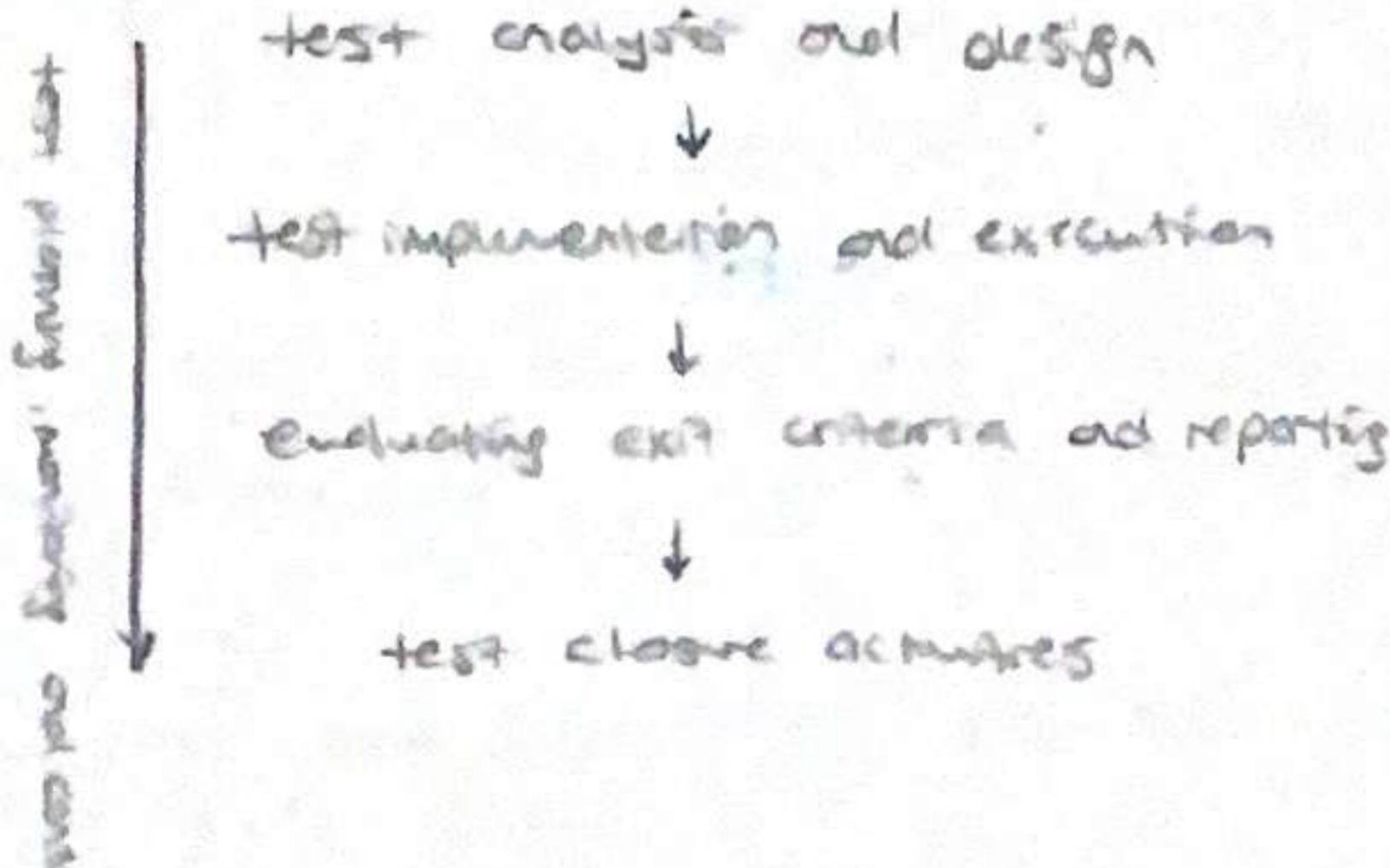
Software testing

- Research process includes verifying that the requirements determined for system or product are met, determine differences and make modifications to correct them
- targets**
 - Detecting errors
 - Confidence about quality
- Providing info for documenting
- Preventing errors

testing principles

- shows the presence of defects, not their absence
- exhaustive testing is impossible
- early testing
- defect clustering
- pesticide paradox
- testing is context dependent
- absence of errors fallacy

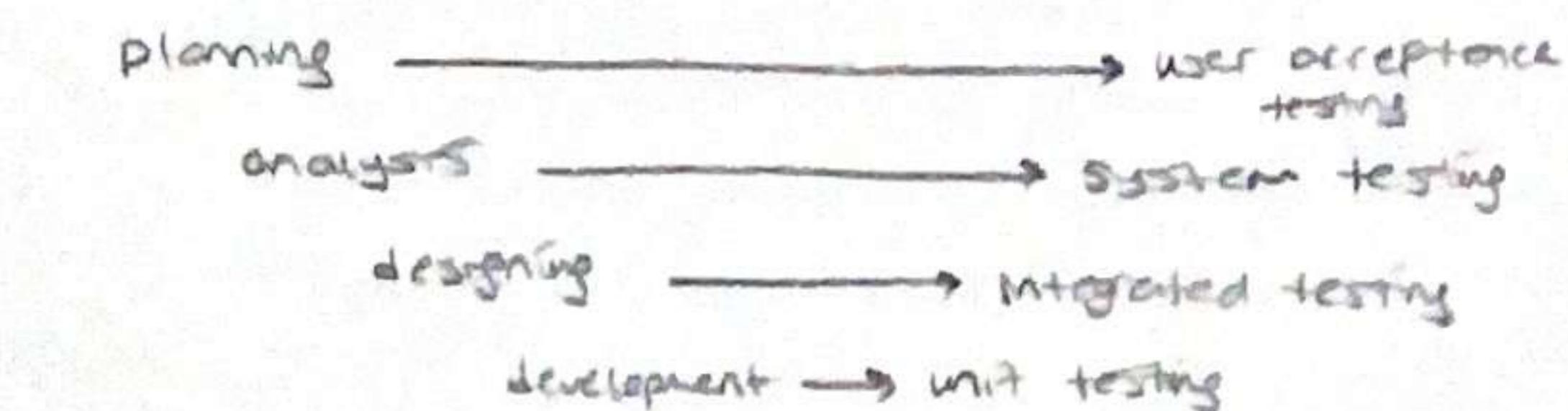
testing process



waterfall model

planning → analysis → designing → development → testing → maintenance

V model

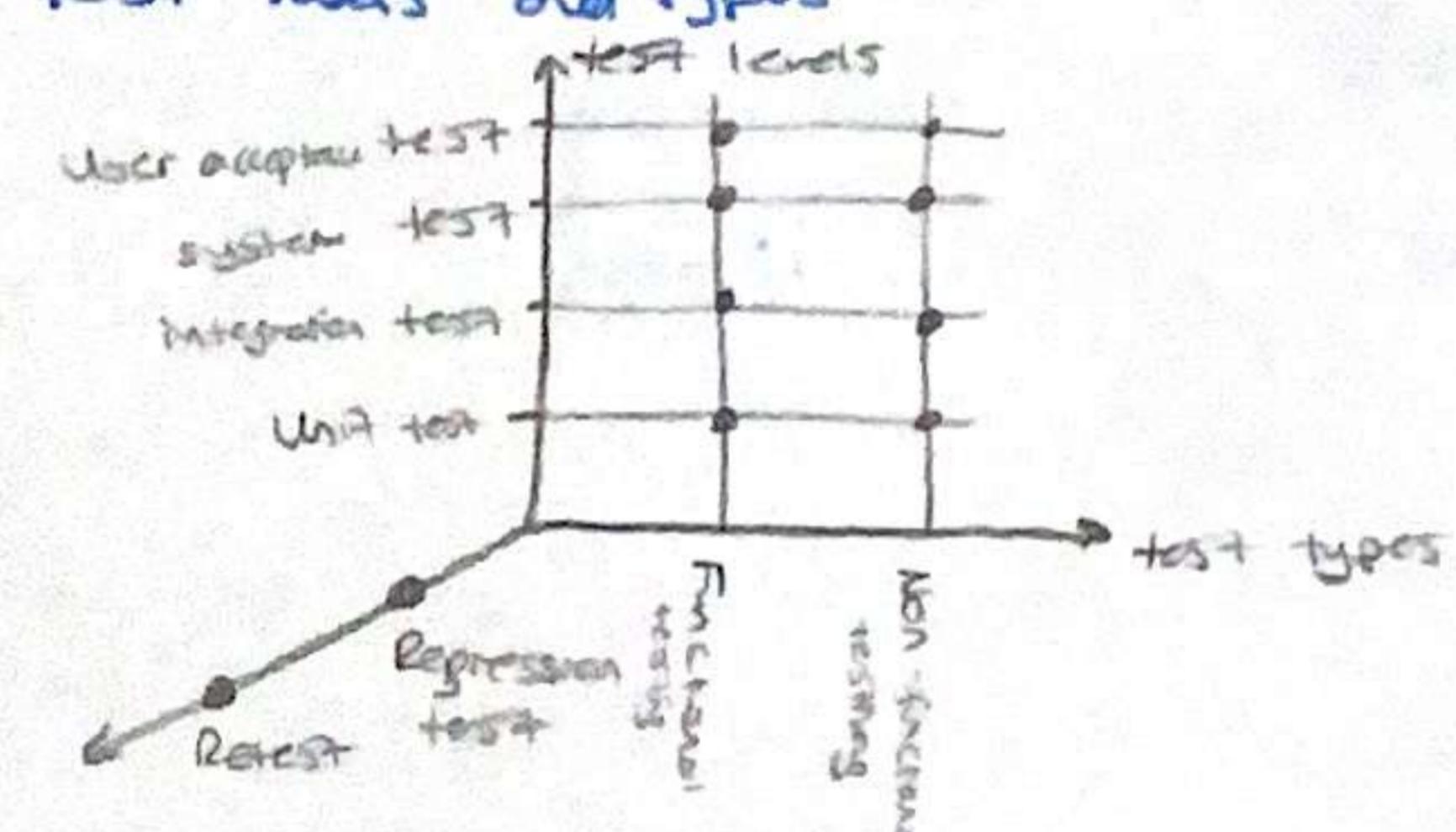


Agile model

Sprint 1 Sprint 2 ... Sprint n

- Planning
- Analysis
- Designing
- Development
- Testing

Test levels and types



Functional testing

- Evaluates whether the software performs the functions specified in the requirements. Focuses on inputs, outputs, user interactions, and system behaviour, including what should do and not do.

Non-functional testing

- Assesses how well the software performs its functions rather than what it does. Measures quality attributes such as performance, usability, reliability, maintainability and portability, often based on standards like ISO 9126.

Re-test

- Test performed to confirm the corrected error

Regression test

- Performed to confirm that corrected error does not cause other effects

Risk

- Impact and probability of situations which may cause adverse consequences if it occurs
- Risk in the software process; represents possible errors which occur when the product is taken into the live environment

Risk identification

- Expert comments
- Risk drafts
- Checklists
- Experiences

$$\text{Risk} = \text{Impact} \times \text{Probability}$$

Impact

- Impression that the error has left on customer, user or other stakeholder
- Loss of reputation, loss of business, loss of trust etc.

Probability

- Probability of an error occurring
- Team conflicts etc.

Risk types

Project risks

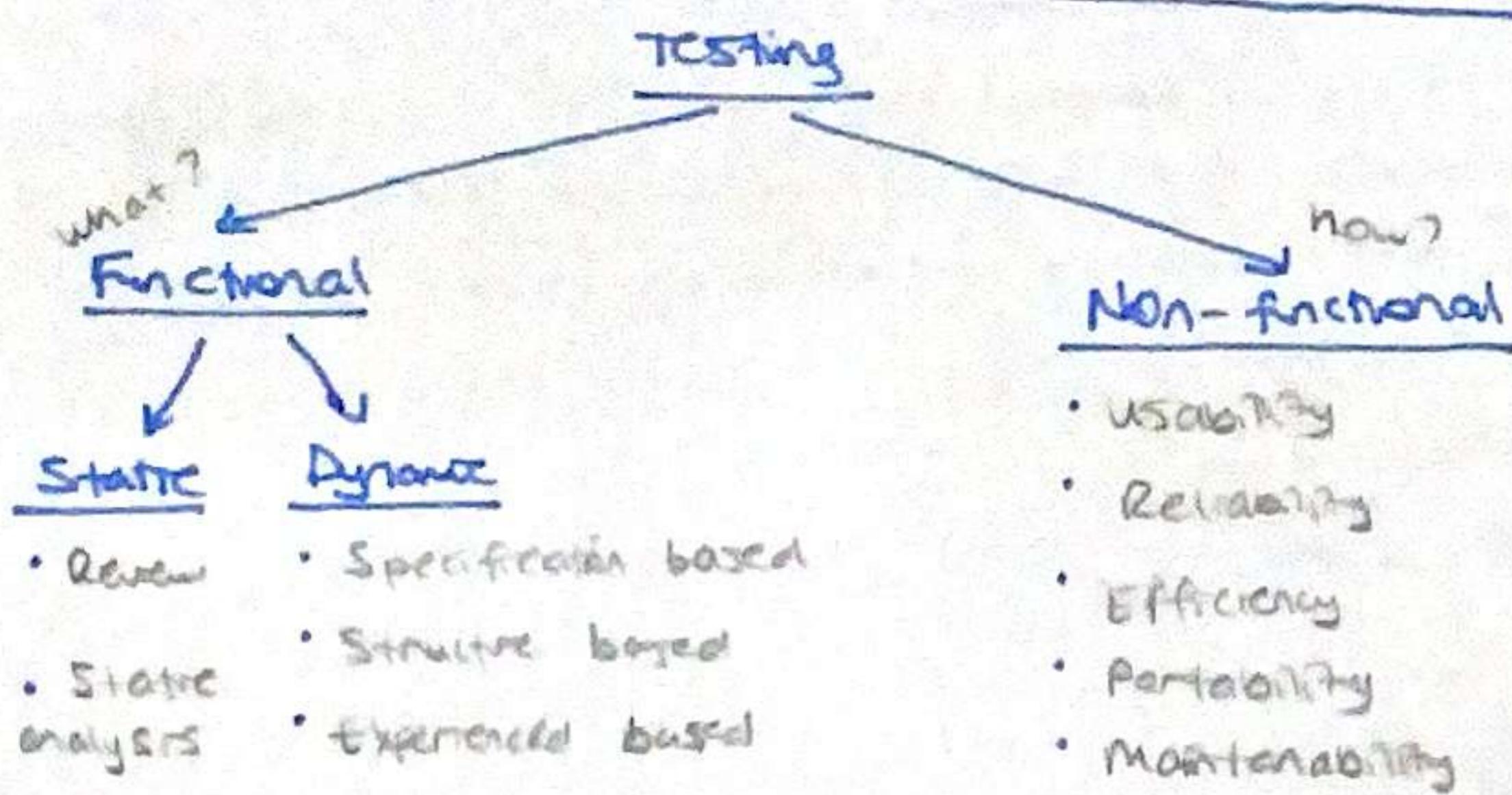
- Not related to software itself, but are problems that may occur in project where software is implemented
- Eg. personal issue, contract issue

Product risks

- Problems with software itself. ISO 9126 quality characteristics are taken into account when determining these risks.
- Eg. functionality, usability, efficiency

Defect report

- Environment
- Scenario
- Actual and expected result
- Data
- Detection time
- Priority
- Severity
- Assigned developer
- Tester



Static testing

- Performed to check defects without executing code

Review

- Informal • walkthrough • Technical review • Inspection

Static analysis

- Reviewing code to find software errors before code is run.

Dynamic testing

- All test performed after the code is compiled and run
 - Depending on type of software, it may differ
- | | | |
|-----------------------|-------------------|--------------------|
| • Specification based | • Structure based | • Experience based |
|-----------------------|-------------------|--------------------|

Equivalence partitioning

- Test design technique divides input data into groups that are expected to behave similarly

Boundary value Analysis (BVA)

- Focuses on testing edge values of input ranges, where defects are most likely to occur. It complements equivalence partitioning by validating both valid and invalid boundaries

Decision table

- Used when system behavior depends on multiple conditions and rules

State transition

- Verifies systems that change behavior based on current state and triggering events. Checks valid and invalid transitions between states

Performance testing

- Measure performance of the system under a certain load and ensure that it reaches desired performance
- Aims to solve the bottlenecks of the system under heavy load with systems such as code and database

Load testing

- Gives info about how much load system is working with maximum performance (how much system can tolerate)

Stress testing

- Uploading system periodically with maximum number of users
- Measure the system's response to such situations in a chaos to determine recovery level when system malfunction is resolved

Reliability testing

Static code analysis

- Data flow anomalies and memory leaks are detected without running the written code
- Based on compliance with coding standards and quality metrics

Penetration testing

- Process of detecting security vulnerabilities with cyber attacks and intrusions using methods that predict malicious attacks, simulating attempts to infiltrate the system with these vulnerabilities and reporting all these transsections

Usability test

- Measure the learnability and satisfaction of the product for the end user

5 components:

- | | | |
|----------------|-------------------|----------------|
| • Learnability | • Memorability | • Satisfaction |
| • Efficiency | • Error tolerance | |

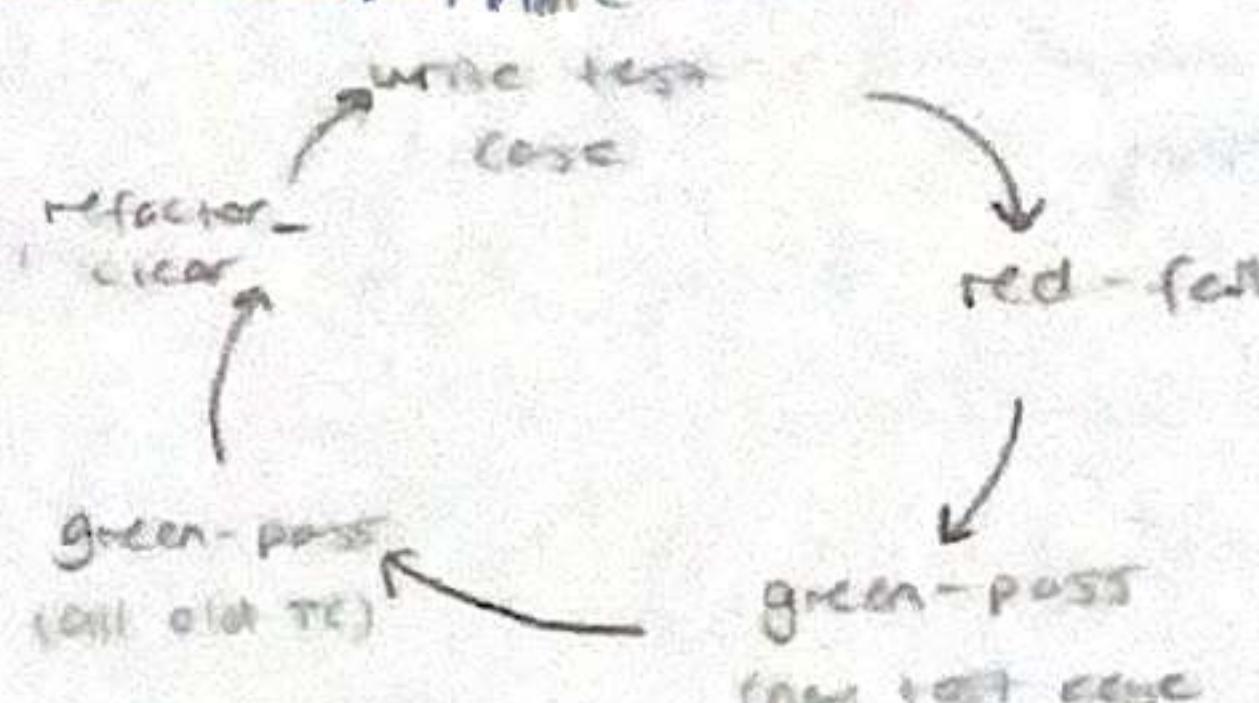
Test data management

- Archiving
- Masking
- Creation
- Clustering
- Protection of analyzing test data requirements, designing test data structures, creating and maintaining test data
- Reliable and real data is masked and provided in a live environment, preventing errors caused by false data and saving time

Test automation

- 1) Period?
- 2) Maturity and maintenance?
- 3) Infrastructure?
- 4) Modularity?
 - 5) The difference between manual and automation
- Unit tests
- Regression tests
- Repetitive tests
- Multi-platform tests

Test-driven development (TDD)



- Development approach where tests are written before code

Behavior-driven development (BDD)

- BDD extends TDD by focusing on business' behavior and user scenarios, often written in natural language, improving collaboration between developers, testers, and stakeholders

6th presentation

Dataum: refers to a single, indivisible unit of information or measurement

Data: represents multiple pieces of information collectively used for analysis and interpretation

Data life cycle

1. Data generation and collection

Data is produced from users, systems, IoT devices and digital platforms

2. Data storage and retention

Data is stored in databases, data warehouses, or cloud platforms

3. Processing and transformation

Raw data is cleaned and transformed via ETL processes

4. Analysis

Analytical and AI tools extract insights, patterns and trends

5. Sharing and usage

Insights are distributed securely to support organizational decisions

State of data management sector

1) Cloud-based

Provide scalable, flexible, cost-efficient infrastructures that support storage, processing, and analytics across distributed environments

2) AI and machine learning integration

Enable faster and more accurate insights, empowering businesses to be more proactive and adaptive

3) Data democratization and self-service analytics

Refers to making data accessible to non-technical users through self-service tools, enabling organization-wide data-driven decision-making

4) Real-time databases

Rapid data processing, streaming data platforms and edge computing are allowing businesses to make non-sensitive decisions with minimal latency

Vector databases

Store numerical embeddings of data and enable semantic similarity search using mathematical operations

Data warehouse (DWH)

- Centralized system designed for analytical processing, optimized for reporting, historical analysis, and decision support rather than transactional operations
- data warehouse → data lake → lakehouse

Large language models (LLM)

- Vector databases: Store numerical embeddings that represent text in a high-dimensional space. Similarity measures such as cosine or dot product are used to quickly retrieve the most relevant documents

Linear algebra

For matrix and tensor operations

Probability theory

For prediction and uncertainty modeling

Calculus

For optimization via gradient descent

ETL (Extract, Transform, Load)

• Extracting

Data from source systems

• Transforming

Into a consistent, quality-controlled format

• Loading

Into a data warehouse or analytics platform

TM presentation

Right: a legally protected interest

Law: the body of rules that regulate society and are enforced by state sanctions

State: a legal entity formed by a politically organised nation or community of nations based on territorial integrity

The constitution

↳ regulates competing rights in working life

• Employer's freedom to work • Right to bodily integrity

• Prohibition of forced labour • Employer's freedom of enterprise

Labour Law (No 4857) → people defined as employees

Turkish Code of Obligations → not defined as "employee"

Labour Contracts Law regulates procedural principles

Principal employer - subcontractor relationship: exists when a principal employer assigns auxiliary or specialized work to another employer, who performs the work at the same workplace using his own employees

Types of employment contracts

Indefinite - Term

→ End date not specified (presumed indefinite if no stated) → Job security provisions applied

Fixed - Term

→ Duration determined, ends upon expiry
→ Cannot be terminated early without just cause
→ Repeated renewals may convert into indefinite
→ Job security provisions do not apply generally

Job security

→ work under an indefinite - term contract
→ Be employed in a workplace with at least 30 employers
→ Have at least 6 months of seniority
→ Not be an employer's representative

Accorded working time

Full-time

→ Weekly: 45 hours
→ Full wages and benefits

Part-time

→ less than two-thirds of full-time hours
→ Rights apply proportionally
→ Does not erode employee rights

Accorded working time

Probationary

→ Max 2 months (4 mon)

→ Termination without notice is possible

→ Wage and insurance obligations remain

Remote work

→ wholly or partially outside of workplace

→ must be in writing

→ OHS obligations continue

Special

On-call work

→ Max 20 hours (less otherwise agreed)

Temporary

→ established as licensed private employment agencies within company groups (both employer responsible for rights)

⇒ Actual working relationship is decisive, not title of contract

Basic principles of labour law - obligations

Employer

- Pay wages
- Treat equally
- Manage and instruct
- Ensure OHS
- Comply with relevant legislation

Employee

- Perform duties diligently
- Act loyally
- Follow instructions
- Avoid unfair competition
- Maintain confidentiality
- Comply with legislation

Fundamental changes in working conditions

→ Notified in writing

→ Approved in writing by employee within 6 business days

Wages

Must be paid in money and may be determined daily, weekly, monthly or per piece

- Gross basic wage
- Net basic wage
- Gross wage including benefits
- Net wage including benefits

Turkish Human Rights Law (Law No. 6701) → equality

Working hours

• Weekly max: 45 hours • Daily max (including overtime): 11 hours

Rest periods

- 15 min for up to 6 hours
- 30 min for 4-7.5 hours
- 1 hour for over 7.5 hours

Overtime (exceeding 45 hours per week)

- Compensated by increased pay or time off (employee's choice)

Leaves

Annual paid: granted after one year of service, may be used in parts

1-5 years → 14 days

15+ years → 26 days

5-15 years → 20 days

18+, 30+ → min 20 days

Public holidays

- Entitled to pay if they do not work
- If they work receive double pay

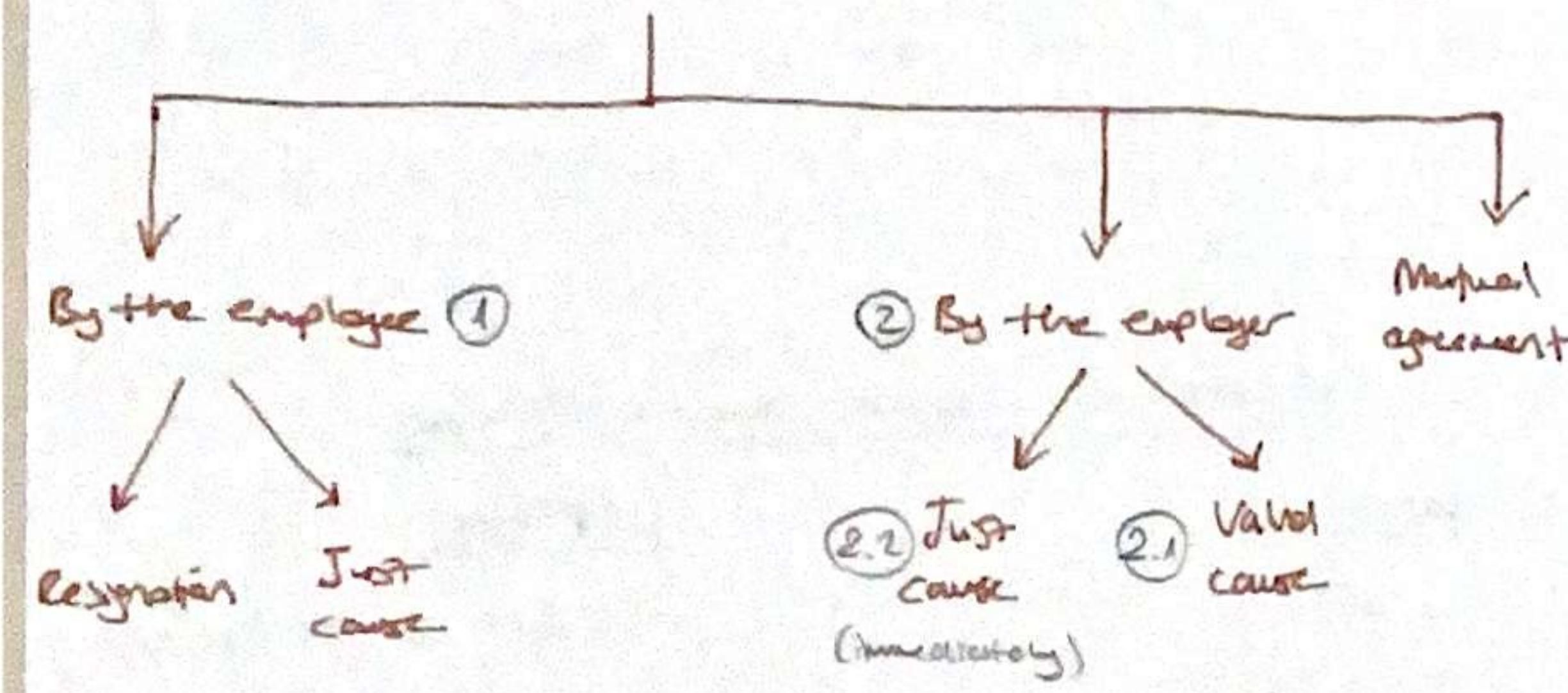
Excuse leave

- Marriage / death of close relatives : 3 days
- BAL of spouse : 5 days

Maternity and sick leave

- Maternity : 8 weeks before and after birth (extended for multiple pregnancies)
- Nursing : 1.5 hours daily until child turns one

Termination of Indefinite Employment Contracts



① Health reasons

- * Cases that do not comply with merit and good faith rules and similar situations → 6 working days after occurred or learned

* Force majeure → cessation of work more than one week

②.1 → Business / workplace / job requirements

→ Employee behavior

→ Employee performance

②.2 → Health reasons

→ Actions violate merit and good faith principles

→ Compelling reasons