Software Engineering

Plan

- Course outline
- Software Engineering
- Motivation
- Sample Reasons
- Best Practices for software development

Outline

Grading Schema

- Final 40%
- Midterm 20%
- Milestones 40%

Milestones

- 3~6 (weight of each varies)
- Covers SE topic
- Deliverables everything & anything (aka written, code, design, … etc.)

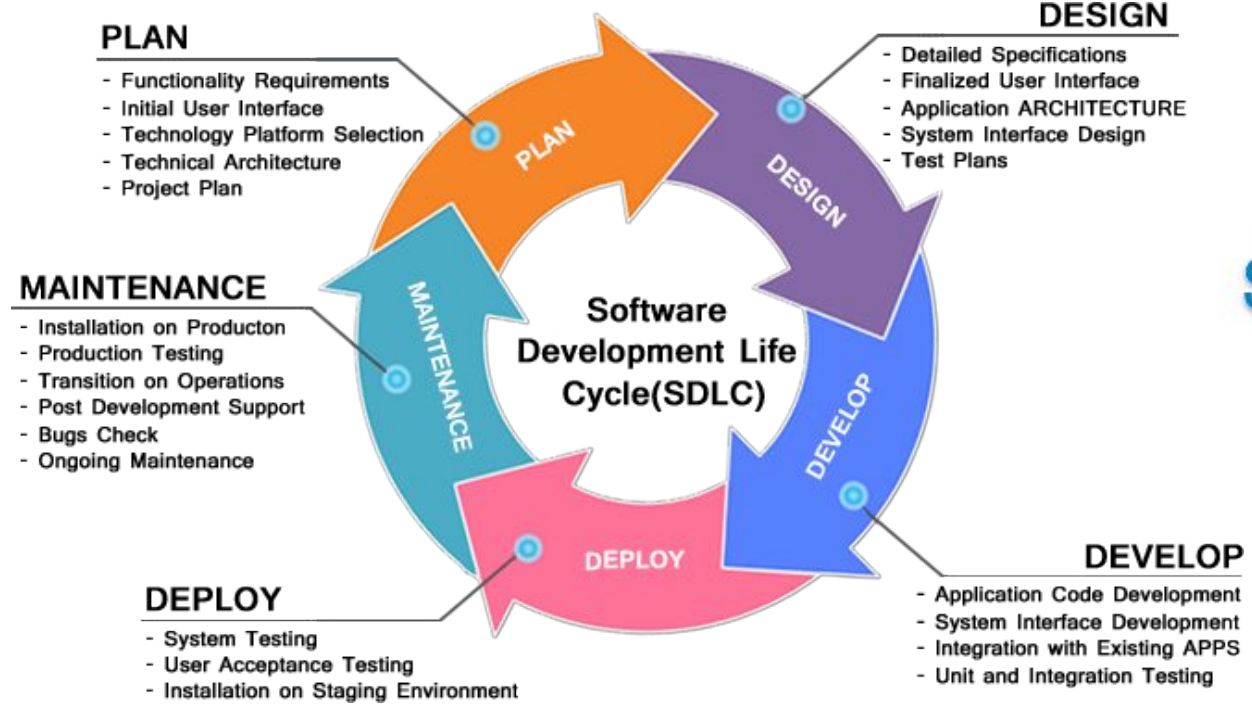
1 Software Engineering

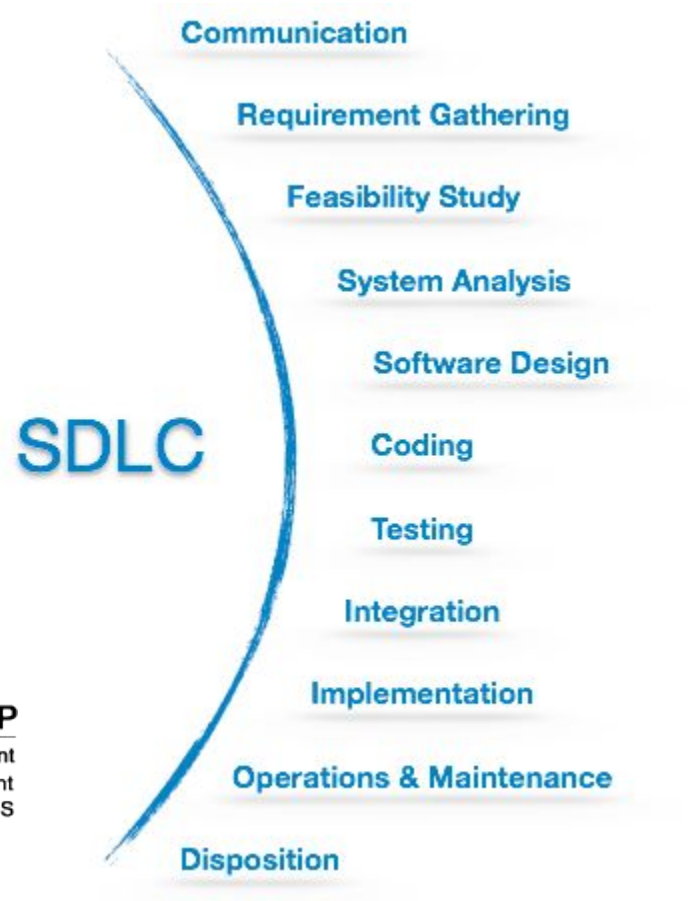
“Software engineering, is the process of analyzing the user’s needs. Then designing, constructing, and testing end user applications that satisfies those needs”

Types of Software

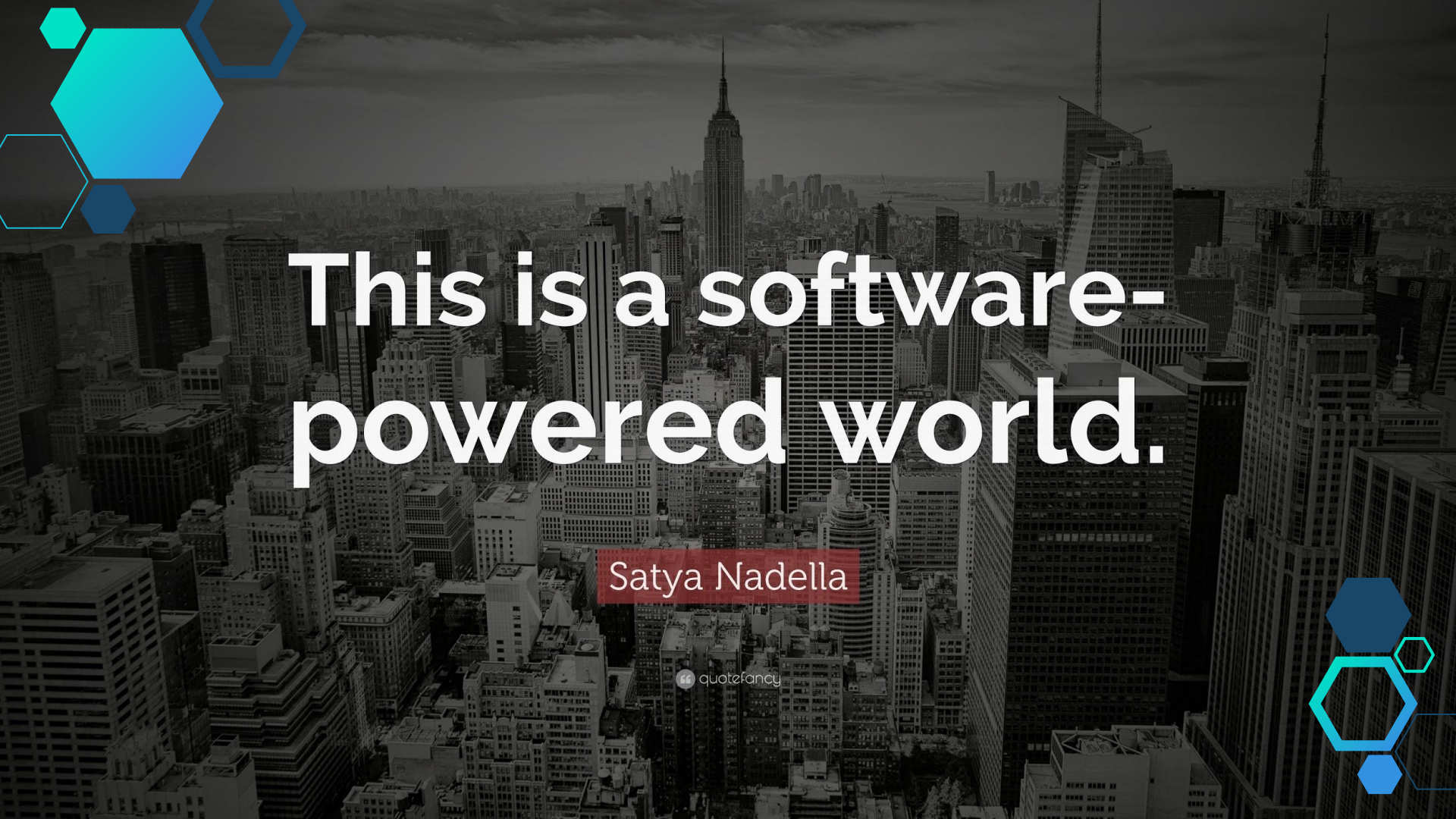
- System
  - Operating System
  - Firmware
  - Utility
  - Device Drivers
  - Programming Language Translators
- Application
  - Word Processors
  - Database Software
  - Multimedia Software
  - Graphics Software
  - Web Browsers

SDLC





2 Motivation


This is a software-powered world.
Satya Nadella
quotefancy

Best Practices for Software Development

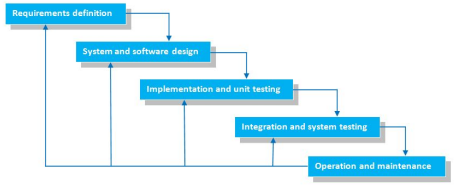
Best practices

- Development process
- Requirements
- Architecture
- Design
- Construction of the code
- Review
- Testing
- Data migration
- Configuration management
- Quality and defects management
- Deployment
- Software Maintenance
- Project management
- Measuring success

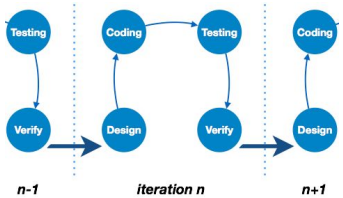
3.01 Development process

Models

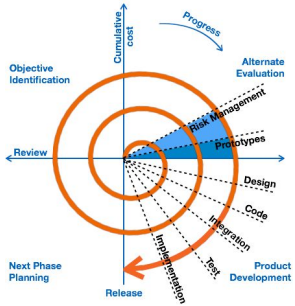
Waterfall



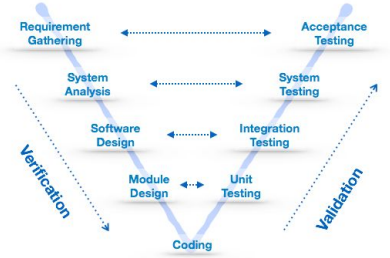
Iterative



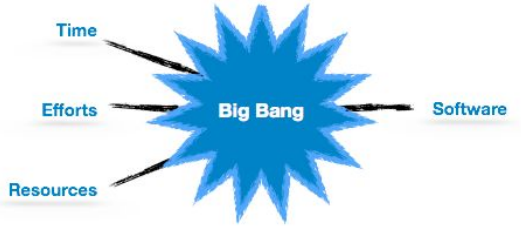
Spiral



V



Big Bang



Methodologies

Agile development

Based on Iterative Model.

Suited for flexibility, continuous improvement, and speed of creating systems.

- Dynamic systems development method (DSDM)
- Kanban
- Scrum

Waterfall development

Based on Waterfall Model.

Suited for simple, unchanging systems .

Spiral development

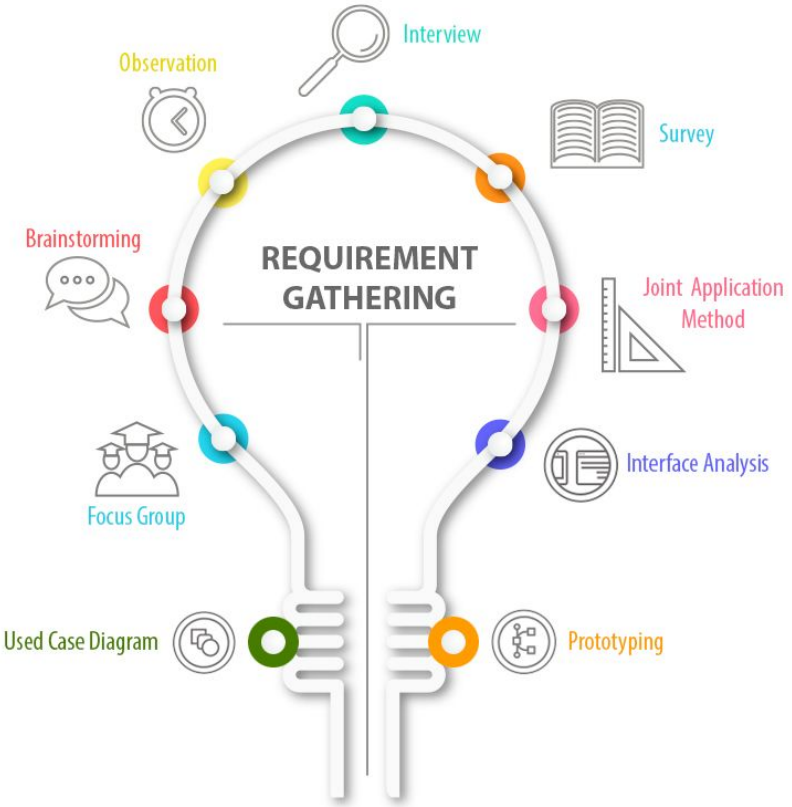
Based on Spiral Model.

Suited to large-scale complex systems.

3.02 Requirements

RG Techniques

- Interviews
- Surveys
- Questionnaires
- Task analysis
- Domain analysis
- Brainstorming
- Prototyping
- Observation



RG Characteristics

- Clear
- Correct
- Consistent
- Coherent
- Comprehensible
- Modifiable
- Verifiable
- Prioritized
- Unambiguous
- Traceable
- Credible source

Software Requirements

Functional, defines what a system is supposed to do.

Example:

- User should be able to mail any report to management.
- Users can be divided into groups and groups can be given separate rights.

Non-Functional, defines how a system is supposed to be.

- Security
- Logging
- Storage
- Configuration
- Performance
- Cost
- Interoperability
- Flexibility
- Disaster recovery
- Accessibility

Architecture & Design

Architecture & Design

Software Architecture

Serves as a blueprint for a system.

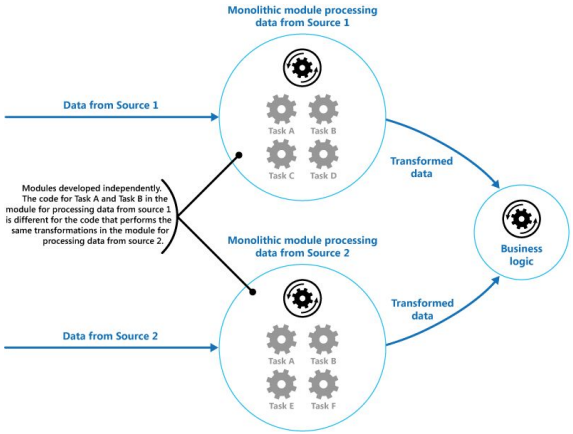
- Business architecture
- Software architecture
- Information architecture
- Information technology (IT) architecture

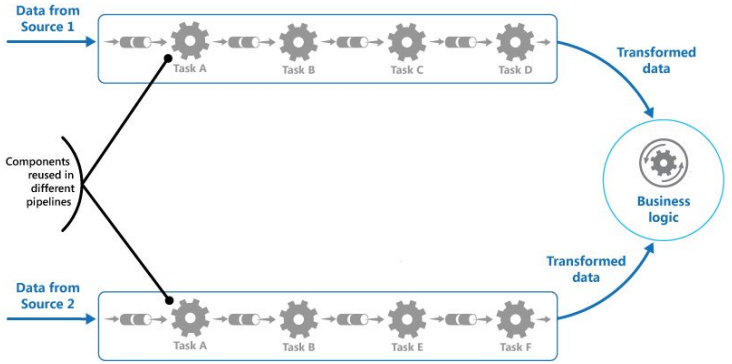
Software Design

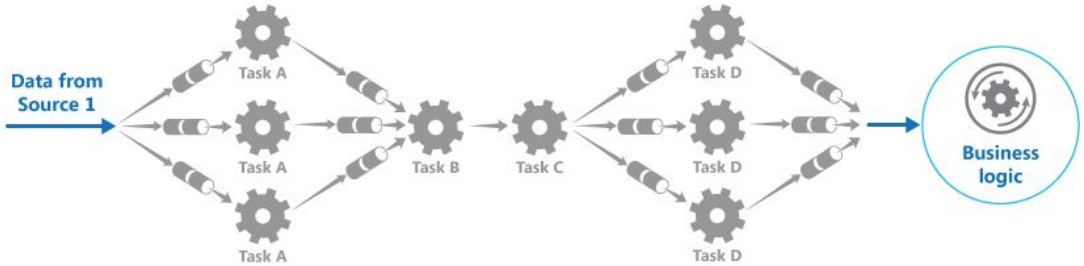
Provides a design plan that describes the elements of a system, how they fit, and work together to fulfill the requirement of the system.

- UML (Unified Modeling Language)
- Architecture View Model (4+1 view model)
- Architecture Description Language (ADL)

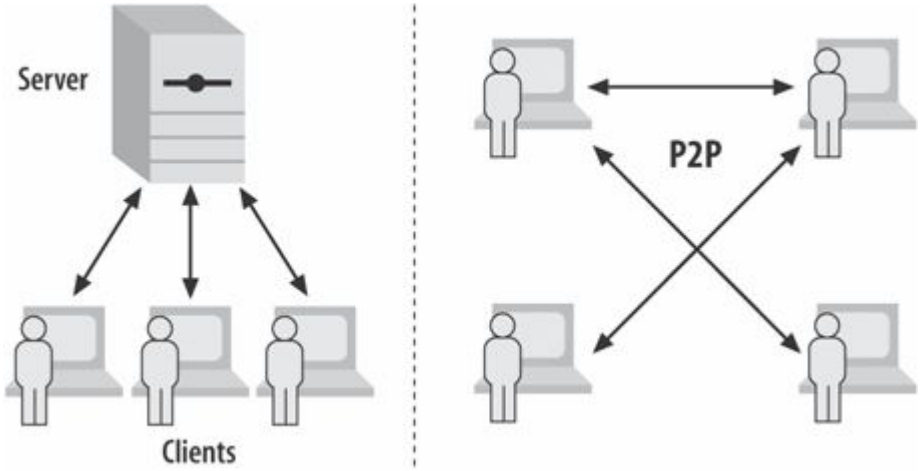
Pipes & Filters



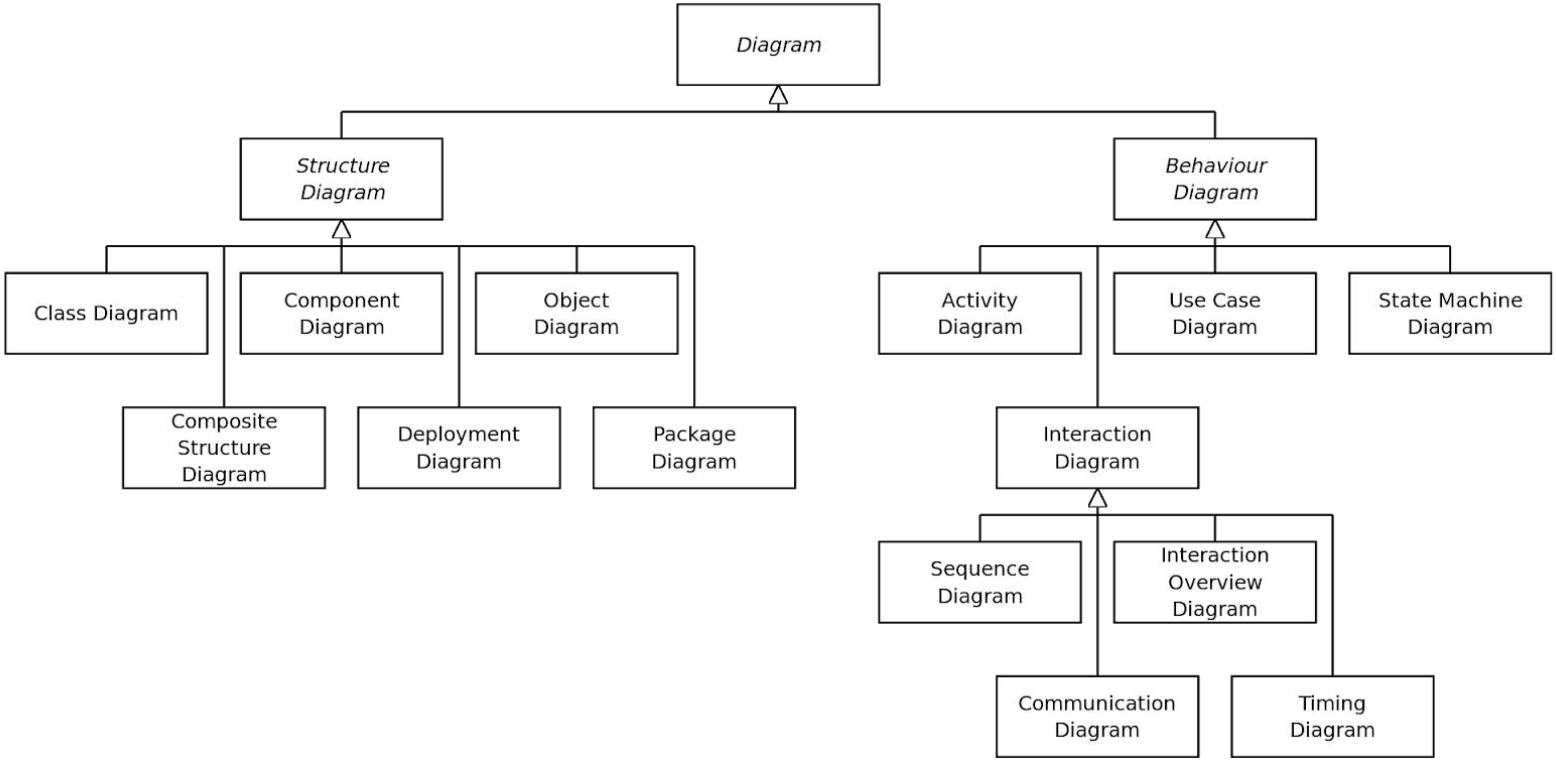




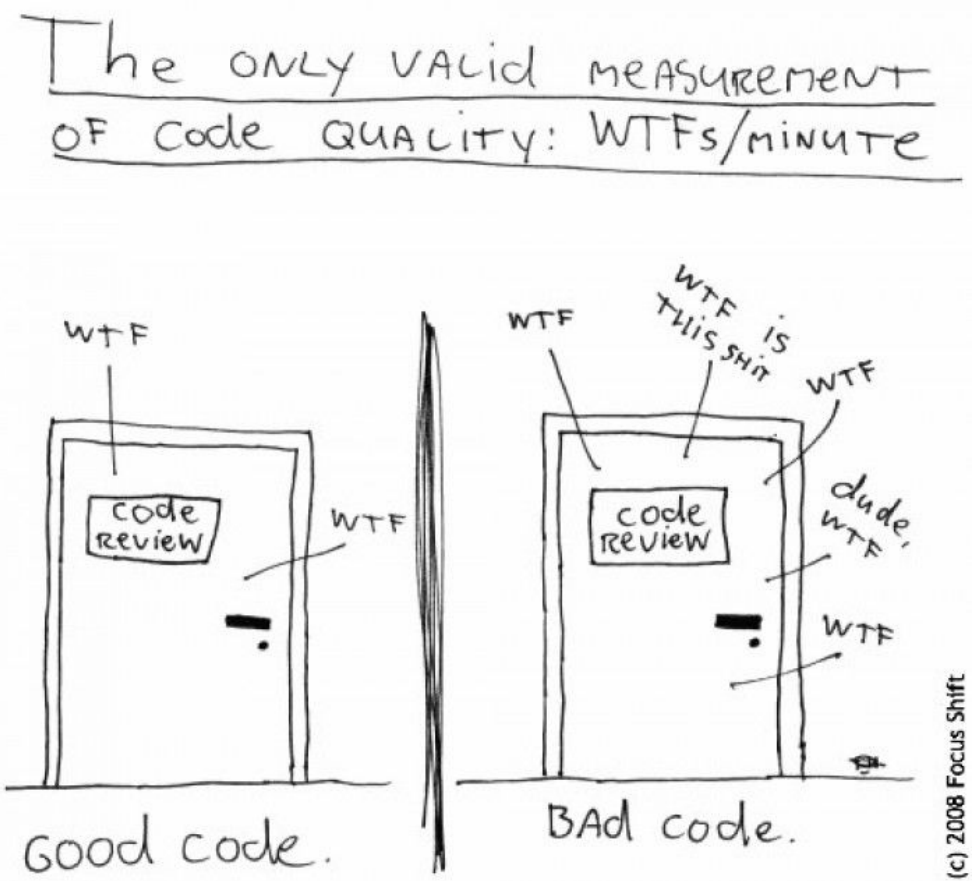
Client-Server
Vs
Peer-to-Peer



Types of UML Diagrams



Construction of the code


The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE
WTF
code review
WTF
Good code.
WTF
WTF IS THIS SHIT
WTF
code review
dude, WTF
WTF
Bad code.
(c) 2008 Focus Shift

Clean Code

- Commenting & Documentation
- Consistent Indentation
- Avoid Obvious Comments
- Code Grouping
- Consistent Naming Scheme
- DRY Principle
- Avoid Deep Nesting
- Limit Line Length
- File and Folder Organization
- Consistent Temporary Names
- Capitalize SQL Special Words
- Separation of Code and Data
- Alternate Syntax Inside Templates
- **Code Refactoring**

3.05 Reviews

Review

- Software Peer Review
  - Code Review
  - Pair Programming
  - Informal
  - Walkthrough
  - Technical Review
  - Inspection
- Software Management Review
- Software Audit Reviews

Testing

Types of Testing

- A/B Testing
- Beta Testing
- Black Box Testing
- Cross Browser Testing
- Exploratory Testing
- Functional Testing
- Load Testing
- Negative Testing
- Nonfunctional Testing
- Pair Testing
- Performance Testing
- Regression Testing
- Security Testing
- Unit Testing
- Usability Testing
- White Box Testing

Data migration

Data Migration Approach

- Analyze source systems
- Review existing documentation
- Gather relevant metadata

- Extract master and transactional data
- Column profiling analysis
- Dependency, uniqueness, redundancy analysis
- Frequency distribution
- Data patterns
- Data quality assessment

- Cleanse data based on business rules
- Parse data
- Match, merge, de-duplication
- Manual cleansing

- Preload reports
- Preload error reports
- Business sign-off

- Load data into target systems
- Exception handling

- Postload reports
- Postload error reports
- Business sign-off

Configuration Management

Configuration Management

Configuration management involves knowing the state of all artifacts that make up your system or project, managing the state of those artifacts, and releasing distinct versions of a system.

Core Patterns:

- Main Line
- Active Development Line
- Workspace Patterns
- Code Line Patterns

Principles:

- Fewer code lines
- Testing
- Integrate early and often
-

Quality and defects management

Quality Assurance Vs Quality Control

- Focuses on processes and procedures rather than conducting actual testing on the system.
- Focuses on actual testing by executing the software with an aim to identify bug/defect through implementation of procedures and process.

Software Quality

- Functional suitability
- Reliability
- Operability
- Performance efficiency
- Security
- Compatibility
- Maintainability
- Transferability
- Maintainability
- Transferability
- Effectiveness
- Efficiency
- Satisfaction
- Safety
- Usability

3.10 Deployment

Deployment

HEROKU

amazon web services™

Azure

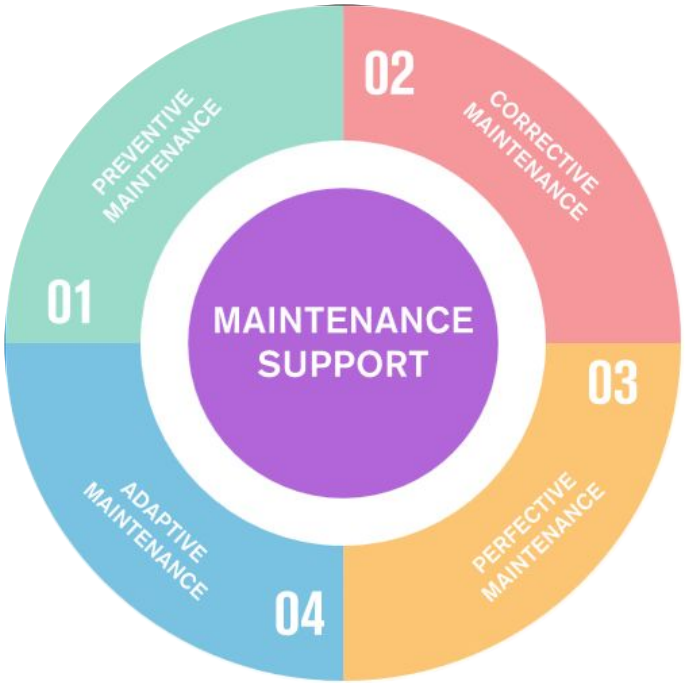
Google Actual Cloud Platform

Software Maintenance

Software Maintenance

Types:

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance



3.12 Project management

Software Project

- Software Project Manager
  - Managing People
  - Managing Project
- Software Management Activities
  - Project Planning
  - Scope Management
  - Project Estimation
- Project Scheduling
- Resource management
- Project Risk Management
- Project Execution & Monitoring
- Project Communication Management
- Configuration Management
- Project Management Tools

3.13 Measuring success

Software Project

- Scope
- Schedule
- Budget
- Team satisfaction
- Customer satisfaction
- Quality

Thanks!

Any questions?

References

- https://www.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html
- https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban
- http://www.professionalqa.com/software-review
- https://www.guru99.com/types-of-software-testing.html
- https://datadominance.wordpress.com/2015/10/21/a-comprehensive-approach-for-data-migration/
- https://dzone.com/refcardz/software-configuration?chapter=1
- https://www.tutorialspoint.com/software_engineering/software_project_management.htm
- https://project-management.com/6-ways-of-measuring-project-success/
- https://docs.microsoft.com/en-us/azure/architecture/patterns/pipes-and-filters
- https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html