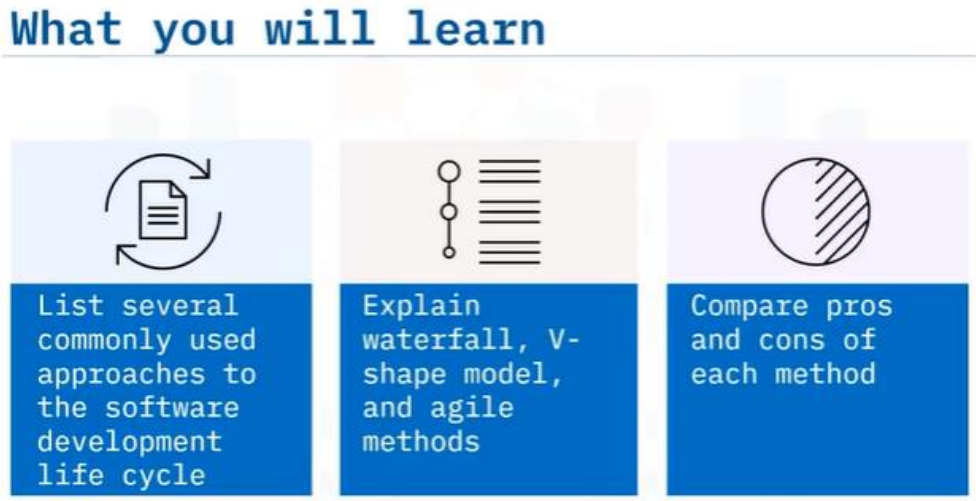
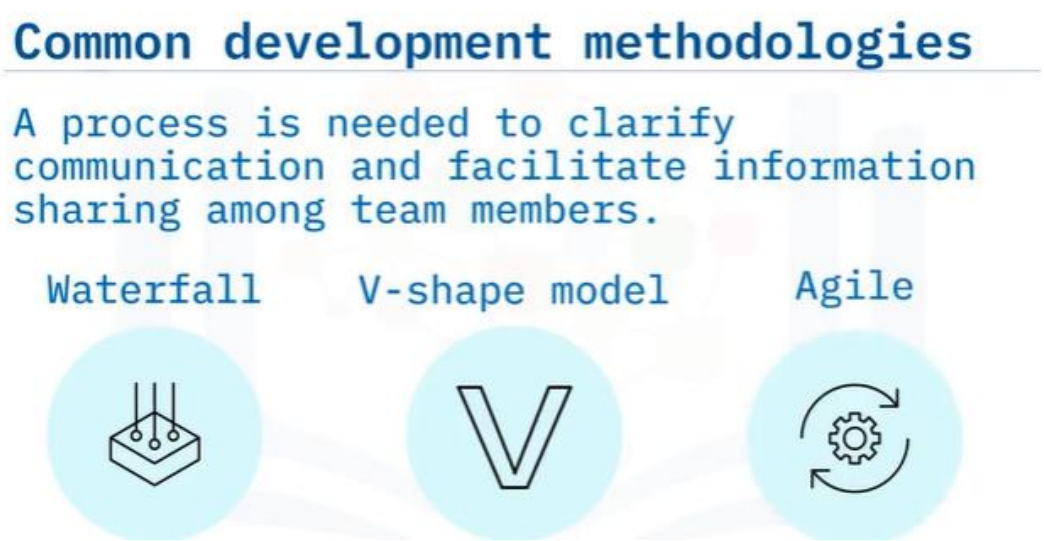
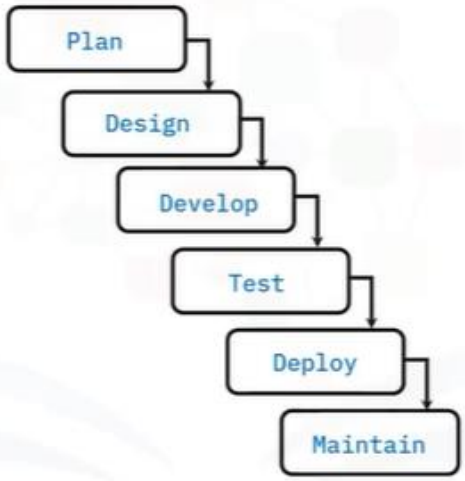
Software development methodologies

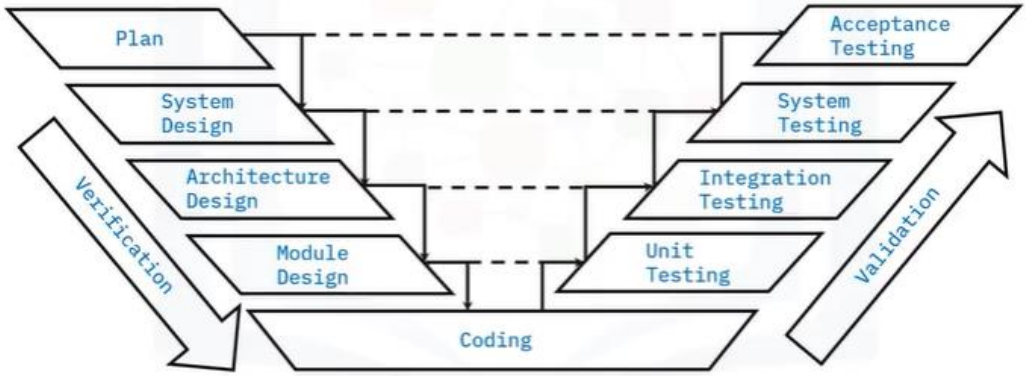




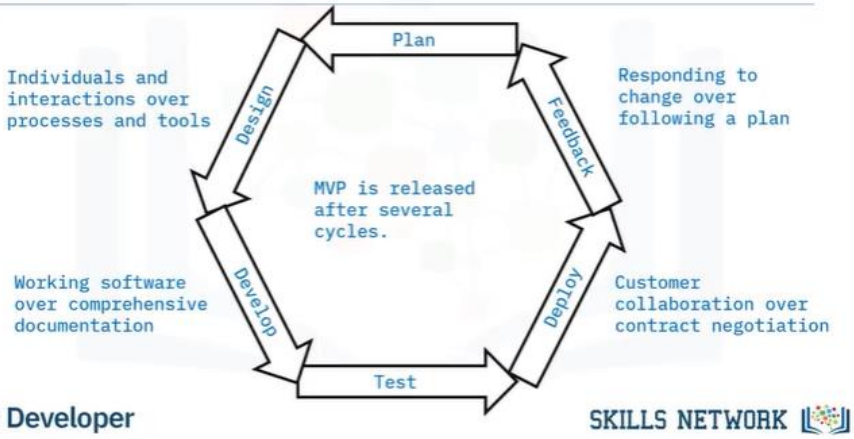
Waterfall method



V-shape model



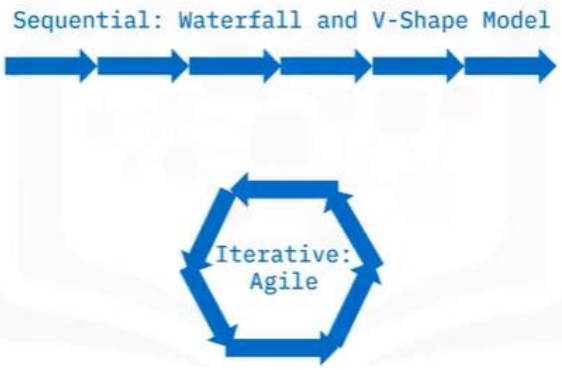
Agile



IBM Developer

SKILLS NETWORK

Sequential vs iterative



Waterfall pros and cons

Pros

- Team members understand their responsibilities due to discrete, well-defined stages
- Easier to estimate budget and allocate resources

Cons

- Lacks flexibility
- Change is hard to accommodate

V-shape model pros and cons

Pros

- Easy to use
- Test plans designed upfront saves development and testing time

Cons

- Rigid
- Does not accommodate changing requirements

Agile pros and cons

Pros

- Changing requirements handled easily
- Feedback incorporated regularly

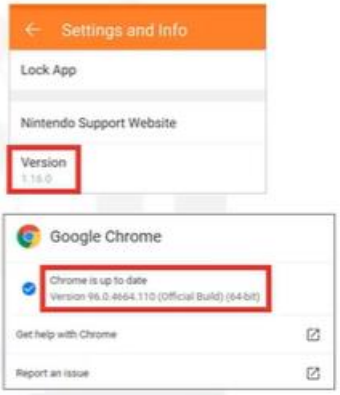
Cons

- Budgeting and resource allocation is challenging

SOFTWARE VERSIONS

Software versions

- Software versions are identified by version numbers
- Version numbers indicate:
  - When the software was released
  - When it was updated
  - If any minor changes or fixes were made to the software
- Software developers use versioning to keep track of new software, updates, and patches



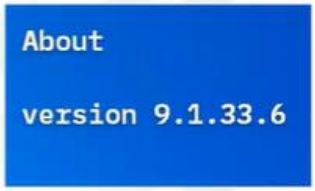
Version numbers

- Version numbers can be short or long, with 2, 3, or 4 sets
- Each number set is divided by a period
- An application with a 1.0 version number indicates the first release
- Software with many releases and updates will have a larger number
- Some use dates for versioning, such as Ubuntu Linux version 18.04.2 released in 2018 April, with a change shown in the third number set

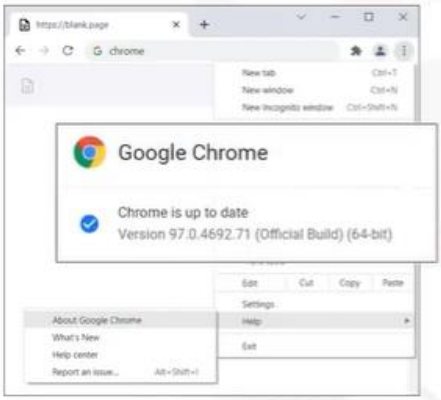
What do version numbers mean?

Some version numbers follow the semantic numbering system and have 4 parts separated by a period

- The first number indicates major changes to the software, such as a new release
- The second number indicates that minor changes were made to a piece of software
- The third number in the version number indicates patches or minor bug fixes
- The fourth number indicates build numbers, build dates, and less significant changes



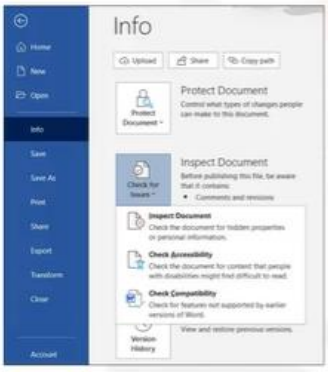
Software version numbers



Practice viewing your software version numbers in a web browser:

- Select the three dots or three lines in the top-right corner of your browser
- Select the menu item Help
- Select About to view the version information
- The version of your web browser will display

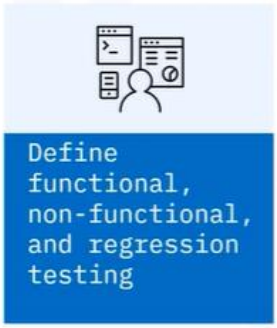
Version compatibility



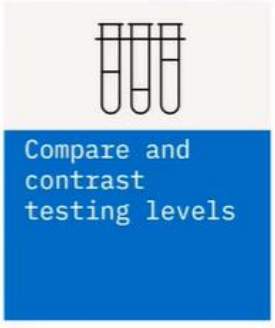
- Older versions may not work as well in newer versions
- Compatibility with old and new versions of software is a common problem
- Troubleshoot compatibility issues by viewing the software version
- Update software to a newer version that is compatible
- Backwards-compatible software functions properly with older versions of files, programs, and systems

SOFTWARE TESTING

What you will learn



Define functional, non-functional, and regression testing

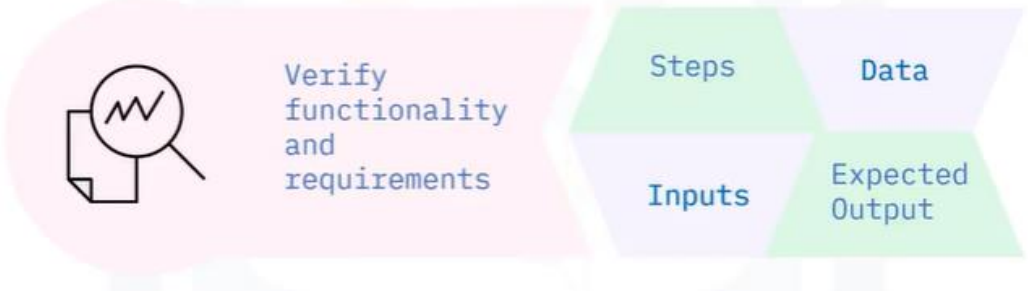


Compare and contrast testing levels

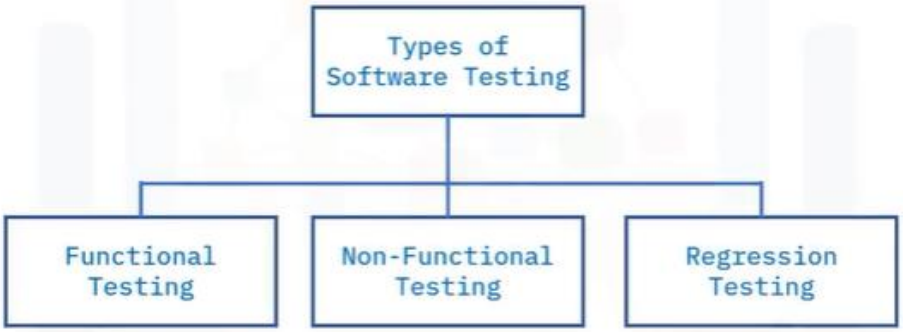
What is software testing?

- Integrate quality checks throughout SDLC
- Purpose
  - Ensure software meets requirements
  - Error-free software

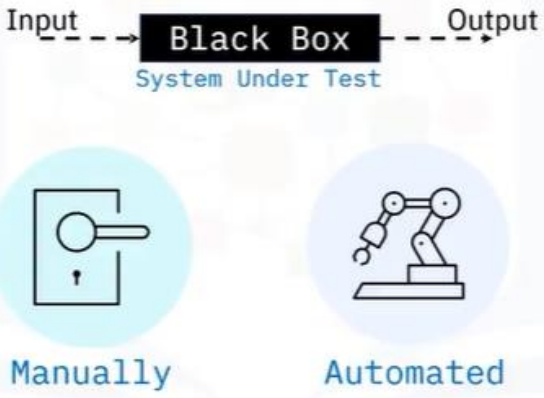
Test cases



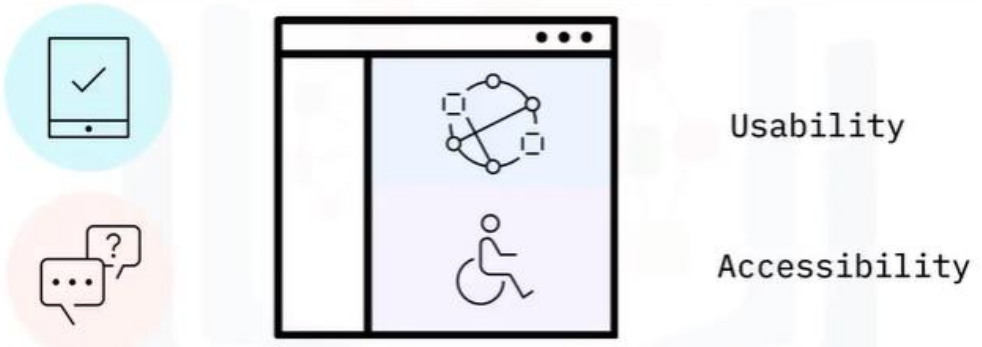
Three types of testing



Functional testing



Purpose of functional testing



Non-functional testing attributes

Performance

Security

Scalability

Availability

Non-functional testing questions

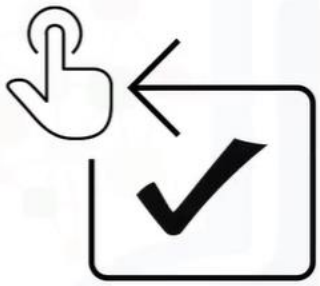
- How does the application behave under stress?
- What happens when many users log in at the same time?
- Are instructions consistent with behavior?

Non-functional testing questions

- How does the application behave under different OSs?
- How does the application handle disaster recovery?
- How secure is the application?

Regression testing

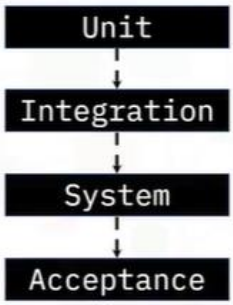
- Confirms changes don't break the application
- Occurs after fixes such as a change in requirements or when defects are fixed



Choosing test cases for regression testing

Choose test cases that contain:

Testing levels



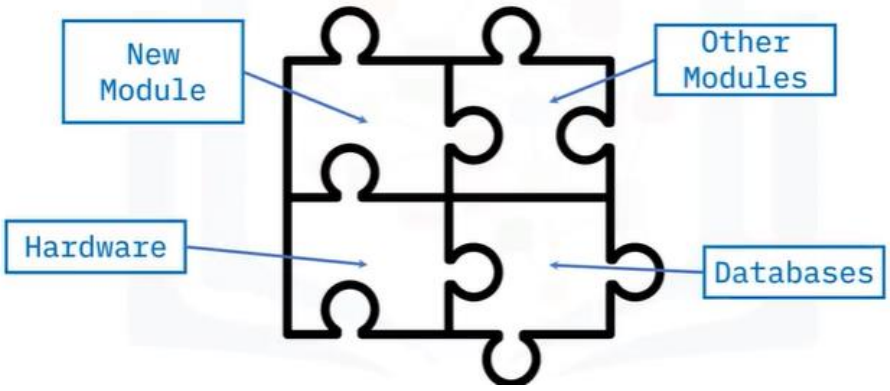
Unit testing

- Test a module of code
- Occurs during the build phase of the SDLC
- Eliminate errors before integration with other modules

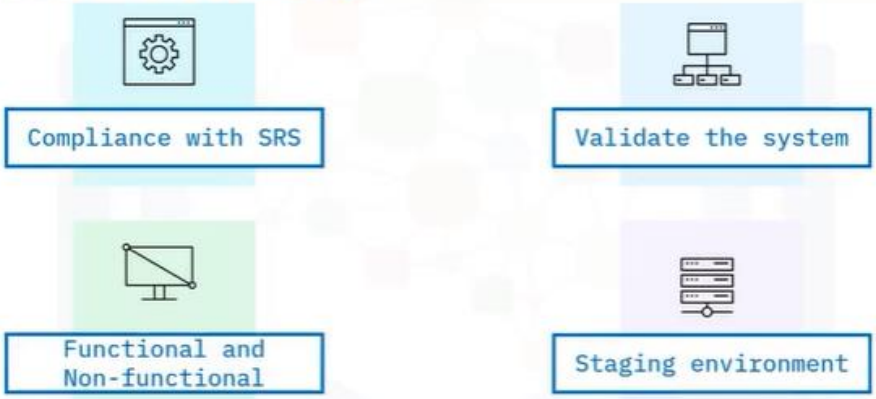
Integration testing

- Identify errors introduced when two or more modules are combined
- Type of black-box test
- Occurs after modules are combined into the larger application

Purpose of integration testing



System testing



Acceptance testing