Acceptance Testing in Agile

- Verifies software meets business requirements and is ready for release.

Acceptance Testing in Agile

“80% of teams in Agile use acceptance tests to align software with business needs.” - Agile Testing Survey

Role of Acceptance Testing

- Acceptance tests define the 'Definition of Done' (DoD) in Agile projects.

Anatomy of Acceptance Testing

Social Media:

Scenario: User creates a new account, uploads a profile picture, and adds friends.

Expected Result: The profile is successfully created and visible to others.

Scenario: User posts a message with an image and tags other users.

Expected Result: The post appears in the user's feed and the feeds of tagged users.

Scenario: User sends a private message to another user.

Expected Result: The message is delivered to the recipient's inbox.

Role in Agile

“Agile teams using Acceptance Test-Driven Development (ATDD) reduce production bugs by 30%.”

Further Reading

https://blog.logrocket.com/product-management/acceptance-test-driven-development/

ATDD & Three Amigos

- ATDD encourages collaboration between developers, testers, and business stakeholders.

ATDD & Three Amigos

Three Amigos meetings

There are three primary representatives involved in a Three Amigos meeting in Agile:

Tester
Discusses software quality concerns and test cases.

Developer
Explains the programming work and anticipated roadblocks.

Business analyst
Defines business requirements and acceptance criteria.

ATDD

The 'Three Amigos' model clarifies requirements before development begins.

ATDD

Acceptance Criteria:

The system should clearly display the classification (e.g., First Class, 2:1, 2:2, Third Class) for each module.

The system should provide a clear explanation of the grade boundaries for each classification.

The system should calculate and display a projected final degree classification based on current grades.

The system should allow me to input predicted grades for future modules to see how they might affect my final classification.

The system should be accessible on multiple devices (e.g., desktop, mobile).

ATDD

Scenario: Student with 70% average across all modules

Expected Result: System displays "First Class" classification.

Scenario: Student with 65% average, but with a failed module

Expected Result: System displays a warning about the failed module and its potential impact on the final classification.

Scenario: Student inputs predicted grades of 60% for remaining modules

Expected Result: System accurately calculates and displays the projected final degree classification based on current and predicted grades.

ATDD

We can confirm ATDD scenario by checking to see it validates against our ‘user stories’ or ‘use cases.’

We can even automate this process.

ATDD

What’s the most common browser resolution(s)?

ATDD

Do our research. Source: StatCounter and BrowserStack

1920x1080: This is the most popular resolution by a significant margin. It's the standard for many modern monitors and laptops, offering a good balance of screen space and sharpness.

360x800(mobile first): This resolution is very common on mobile devices, especially in the Android ecosystem. It reflects the shift towards taller, narrower screens on smartphones.

1366x768: This was once extremely popular for laptops and some smaller monitors. While it's less common on newer devices, it still holds a significant share due to the large number of devices with this resolution still in use.

ATDD

Now what?

ATDD


```
function resizeToResolution(width, height) {
    window.resizeTo(width, height);
  }

  // Example usage:
  resizeToResolution(1920, 1080); // Resize to 1920x1080
  resizeToResolution(360, 800);  // Resize to 360x800
  resizeToResolution(1366, 768); // Resize to 1366x768
```

Selenium for Automated Testing

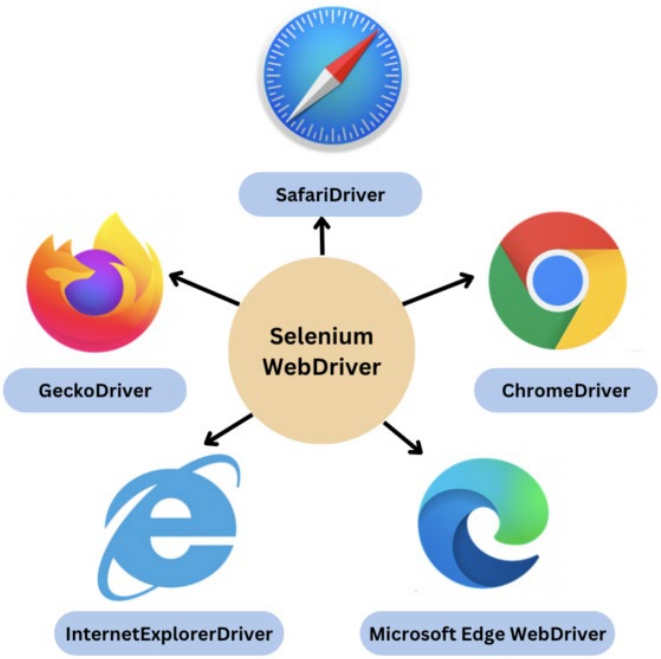
- Selenium automates web browser interactions to conduct end-to-end tests.

Selenium Use

“75% of web automation frameworks rely on Selenium for testing workflows and user interfaces.”

Selenium in Acceptance Testing

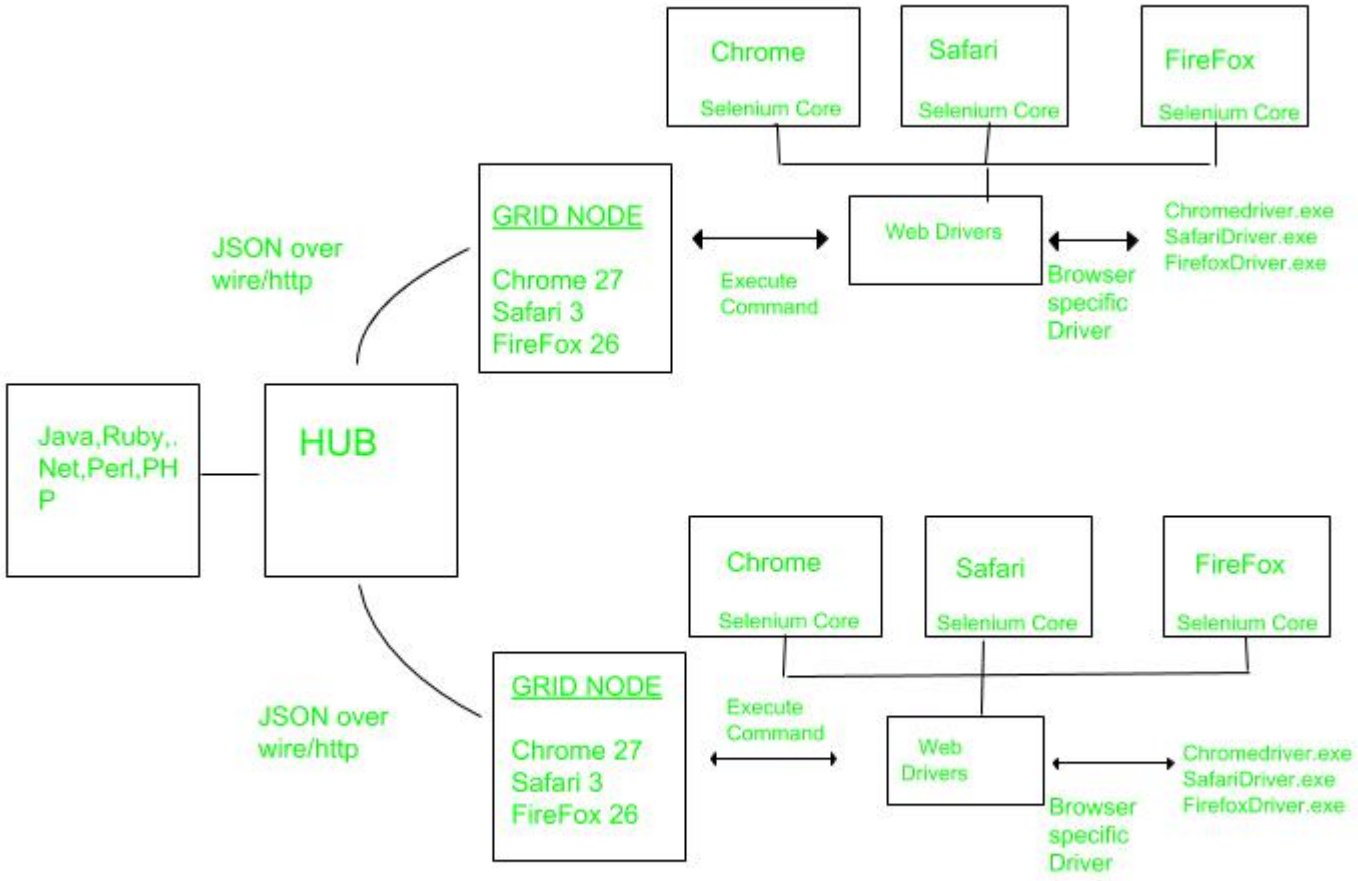
- Automates acceptance tests to verify workflows such as login, navigation, and transactions.



Selenium in CI Pipelines

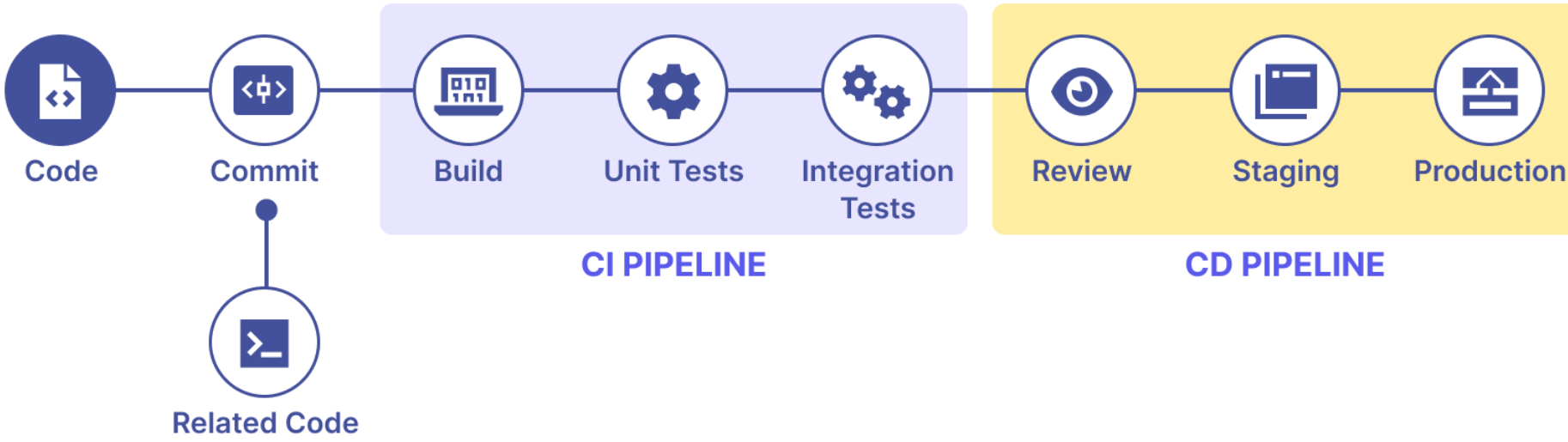
- Selenium tests can be integrated into Continuous Integration pipelines to provide quick feedback on code changes.

Selenium in CI Pipelines



Selenium in CI

“Using Selenium in CI pipelines reduces testing time by 40%.”



W3C Validator

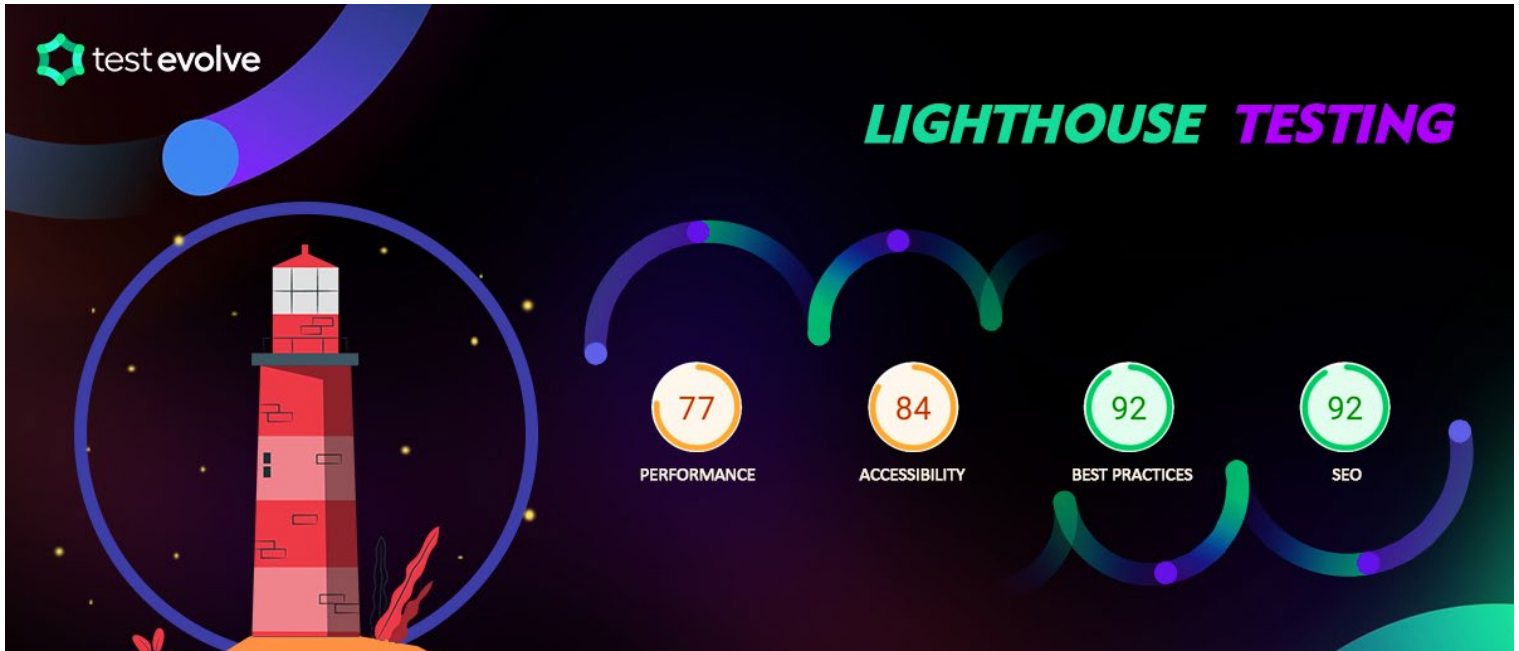
- The W3C Validator checks HTML and CSS code for compliance with web standards, improving cross-browser compatibility.

W3C Validator Use

Validating code reduces cross-browser issues and improves accessibility by 20%.

Lighthouse for Web Testing

- Lighthouse is an open-source tool that assesses web performance, accessibility, SEO, and best practices.



Lighthouse Impact

“Websites with high Lighthouse scores rank 30% better in Google search results.”

Google will go as far as to **not index** your site if you do not follow best practice!

Other Testing Tools

- Tools like Pa11y and Axe help automate accessibility checks to ensure compliance with web accessibility standards.

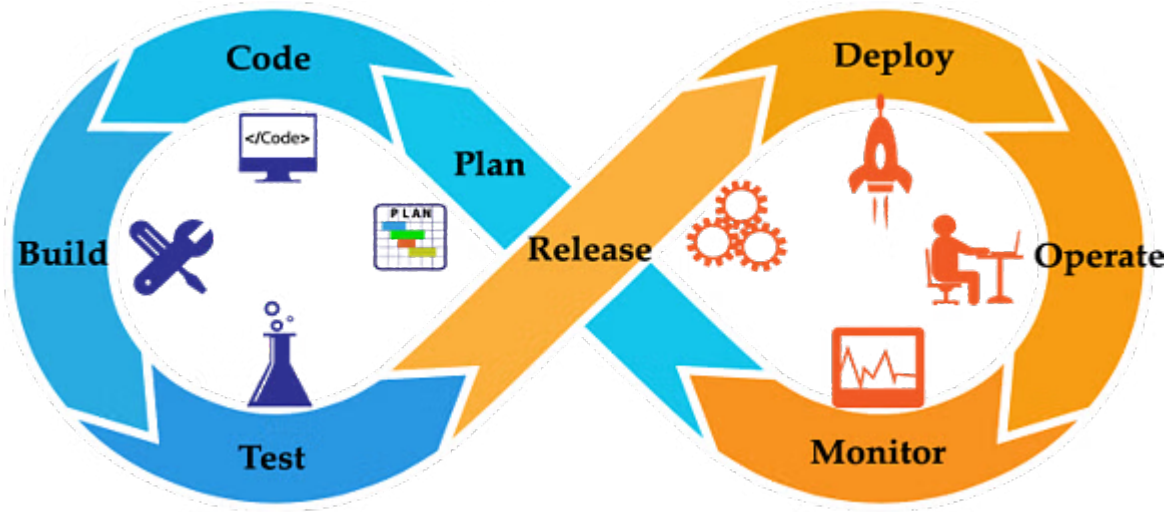
Accessibility Tools

“Axe-core is used by over 60% of accessibility testers worldwide.”

https://github.com/dequelabs/axe-core

Continuous Integration & Testing

- CI integrates code changes frequently and runs automated tests to ensure quality in each iteration.



CI & Testing Impact

“Automating tests in CI pipelines can reduce deployment times by 50%.”

Integrating Testing Tools

- Jenkins, GitHub Actions, and other CI tools allow the automation of tests like Selenium, W3C Validator, and Lighthouse.

CI Tooling

70% of high-performing teams integrate automated tests into their CI pipelines.

```
// Function to validate an email address
function isValidEmail(email) {
    const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return re.test(email);
}

// Test cases
console.assert(isValidEmail("test@example.com"), "Valid email should pass");
console.assert(!isValidEmail("test@example"), "Invalid email should fail");
```

Performance & Load Testing

- Tests how the application performs under heavy load and stress to ensure it can handle real-world traffic.

Performance Testing Example

Example: Stress testing an app with over 1,000 concurrent users to ensure stability. You could also look at time to execute e.g.

```
const start = performance.now();
for (let i = 0; i < 1000000; i++) {
  // Do something
}
const end = performance.now();
console.log(`Execution time: ${end - start} ms`);
```

Gatling for Load Testing

- Gatling is an open-source load and performance testing tool that helps identify bottlenecks.

Gatling Impact

“Using Gatling improves app scalability by 25% by detecting bottlenecks early.”

Gatling Impact

```
import scala.concurrent.duration._

import io.gatling.core.Predef._
import io.gatling.http.Predef._

class BottleneckSimulation extends Simulation {

  val httpProtocol = http
    .baseUrl("https://your-api-endpoint.com") // Replace with your API endpoint
    .acceptHeader("application/json")

  val scn = scenario("Bottleneck Test")
    .exec(http("Get Data")
      .get("/data")
      .check(status.is(200)))
    .pause(1) // Adjust pause to simulate user behavior

  setUp(
    scn.inject(
      rampUsers(1000) during (1 minute) // Adjust user count and ramp-up time
    ).protocols(httpProtocol)
  )
}
```

Security Testing

- Tools like OWASP ZAP automate security testing, helping detect vulnerabilities in web applications.

  https://github.com/zaproxy/zaproxy

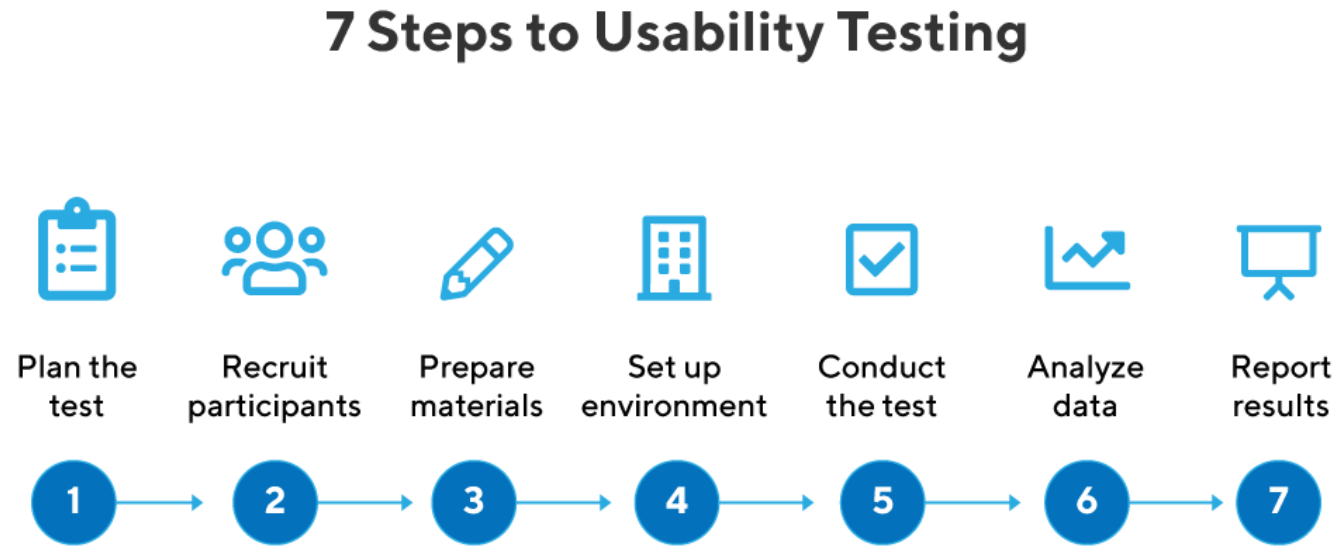
Security Impact

“98% of websites that pass OWASP ZAP security checks avoid common vulnerabilities.”

Most penetration testers will focus heavily on the “low hanging fruit.” Generally safe is, for the most, part safe.

Usability & Accessibility Testing

- Usability and accessibility tests ensure that applications are user-friendly and meet accessibility standards.



Accessibility Impact

Accessible websites see a 25% increase in user engagement due to inclusive design.

The Principles of Universal Design

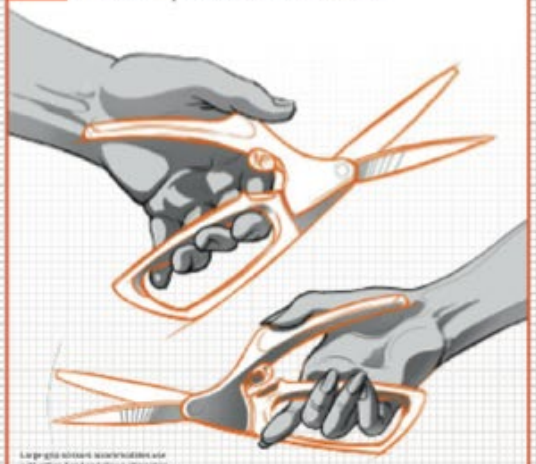
The design of products and environments to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design.

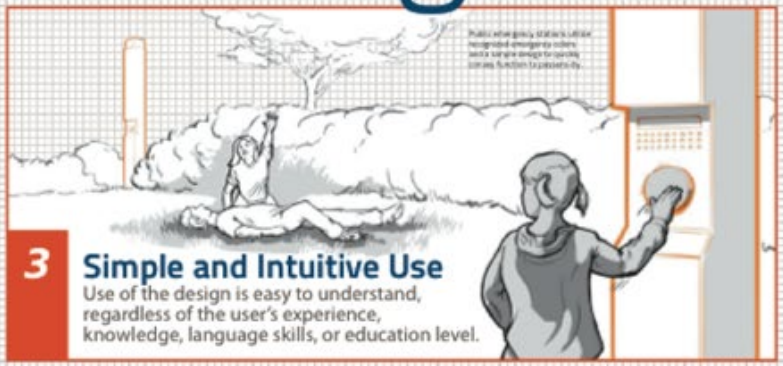
1 Equitable Use

The design is useful and marketable to people with diverse abilities.

2 Flexibility in Use

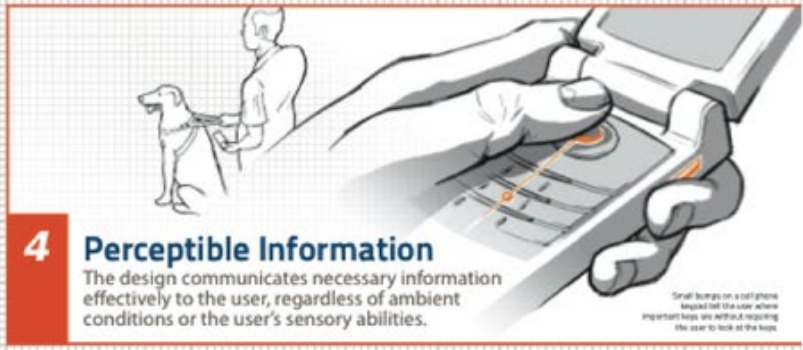
The design accommodates a wide range of individual preferences and abilities.





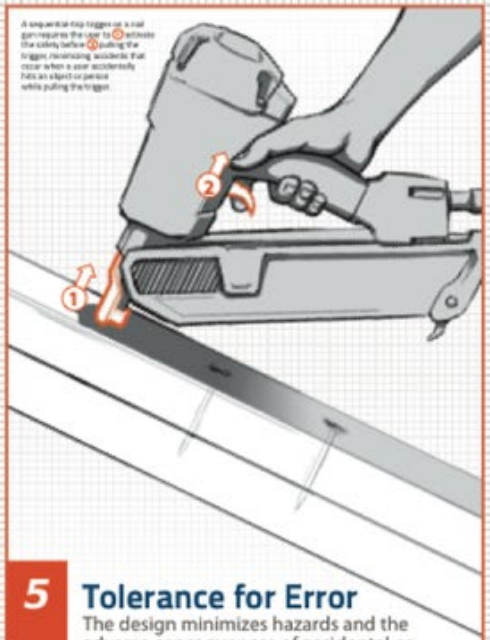
3 Simple and Intuitive Use

Use of the design is easy to understand, regardless of the user's experience, knowledge, language skills, or education level.



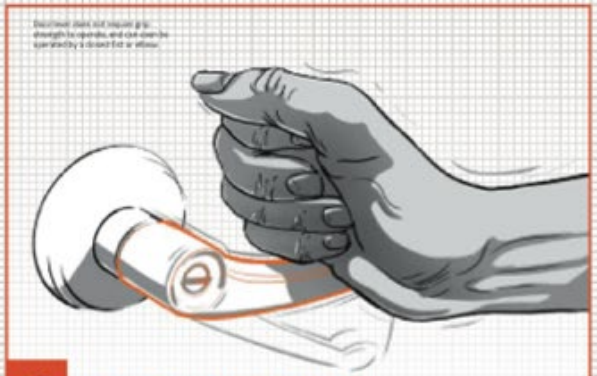
4 Perceptible Information

The design communicates necessary information effectively to the user, regardless of ambient conditions or the user's sensory abilities.



5 Tolerance for Error

The design minimizes hazards and the adverse consequences of accidental or unintended actions.



6 Low Physical Effort

The design can be used efficiently and comfortably and with a minimum of fatigue.

7 Size and Space for Approach and Use

Appropriate size and space is provided for approach, reach, manipulation, and use regardless of user's body size, posture, or mobility.

Center for Universal Design at NC State

Web: design.ncsu.edu/cud

E-mail: cud@ncsu.edu