



**Copado Robotic Testing (CRT)** 

Implementation

Theory & Practice Section 1



# **Theory & Practice**

Everything you need to know and learn doing





# **Introduction to Testing**



# **Software Testing**

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs and improving performance.



# **Business vs. IT Testing Perspectives**

## **Business Requirements**

- High-level business needs
- Connected with the business problem or business opportunity
- Broad and high level
- Written from the point of view of the client

## **Functional Requirements**

- Functions required to fulfill the business needs
- Connected with the solution or the software
- Specific and detailed
- Written from the point of view of the system/application

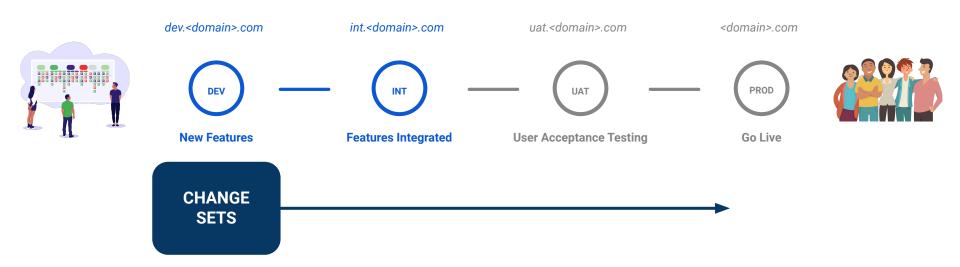
## **Non-Functional Requirements**

- Non-functional Requirements (NFRs) define 'how well' systems do what they do
- This includes characteristics such as their performance, security, maintainability, scalability, and ease of use



# **Software Delivery**

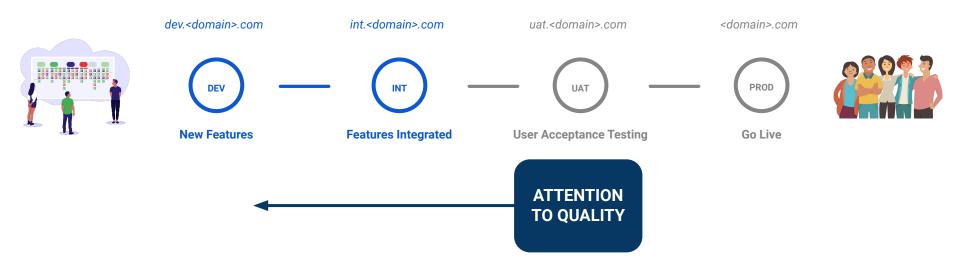
is the complete set of processes required to design, build, test, deploy and manage software





# What does Shifting Left Testing mean?

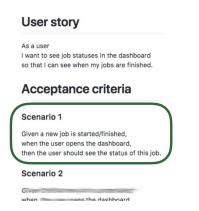
Start testing early in software development stages





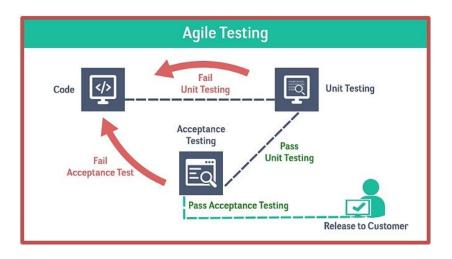
# **Agile Testing**

## Means pro-actively engage with testing of new features starting at the design





Agile Testing already starts with defining test cases while refining user stories.



**New vs. Existing** 

Copado Robotic Testing allows you to run tests in Development or Regression Mode.



Copado Robotic Testing lets you target multiple environments.



SPRINT CLOSED

**SPRINT START** 

# **Manual Testing**

## **Pros**

- Human involvement
- Instantly executable
- Flexible
- For anyone

## Cons

- Manual activity
- Slow test execution
- Repetitive activity
- Time consuming
- Costly activity
- Boring activity
- Growing activity
- Sensitive to human error
- Manual reports & distribution



# **Automated Testing**

## **Pros**

- Repeatable (Save time and Money)
- Easy to plan and schedule
- Thorough and Consistent
- Fast Test Executions
- Frequent Test Executions
- Generated Reports
- Automatic Report Distribution

## Cons

- Investment necessary
- Requires onboarding / training
- Requires ownership
- Maintenance required
- Not everything is easy to automate



# **Functional Testing Types**

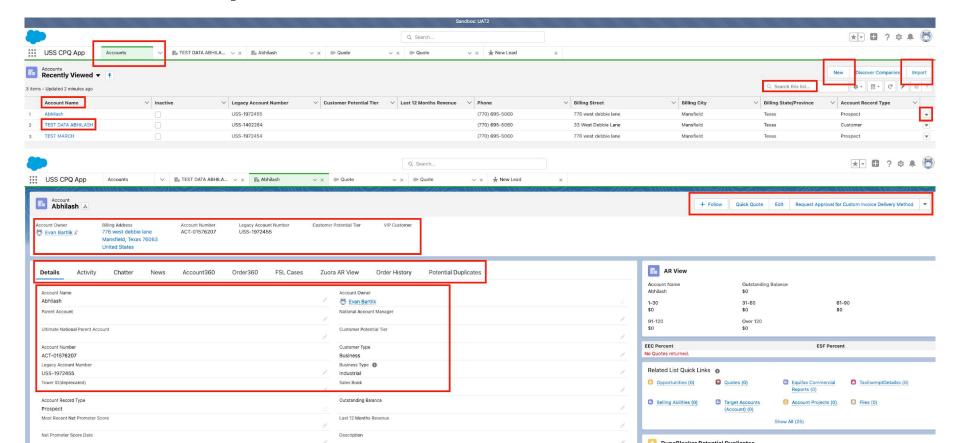
	Туре	Who	When	Requirements	Characteristics
1	Unit Testing / Apex tests (out of scope CRT)	Developers	Dev Int UAT Prod	Functional	<ul> <li>Executed at code level</li> <li>Runs as part of the software build process</li> <li>Tests units of code</li> </ul>
2	Component Testing	Developers Testers	Dev Int UAT	Functional	<ul> <li>Did we build working components?</li> <li>Performed after unit testing</li> <li>Executed at application level (post build and deployment)</li> <li>Testing individual components</li> </ul>
3	Integration Testing	Developers Testers	Dev Int UAT	Functional	<ul> <li>Did we build working components that interact?</li> <li>Performed at application level after system / component testing</li> <li>Ensuring various components within a program or system can function together well</li> </ul>
4	Acceptance Testing  Functional Acceptance Testing Business Acceptance Testing	Developers Testers Business Users	Dev Int UAT	Functional Business	<ul> <li>Did we build the right thing? Is this what the customer needs?</li> <li>Focuses on the client's use of the system and how it functions as a whole</li> <li>Acceptance tests are one of the last tests types that are performed on software prior to a promotion.</li> <li>Acceptance testing is used to determine whether a piece of software satisfies all of the requirements from the business or user's perspective</li> <li>Suitable for end-to-end testing, testing business processes</li> </ul>
5	Regression Testing	Testers	Dev Int UAT	Functional Business	<ul> <li>Regression testing is a style of testing that focuses on retesting after changes are made</li> <li>Used for result comparison between releases</li> <li>Used to obtain an increasing level of confidence releasing software</li> <li>Larger regression suites do not provide quick feedback</li> </ul>

# **Non-Functional Testing**

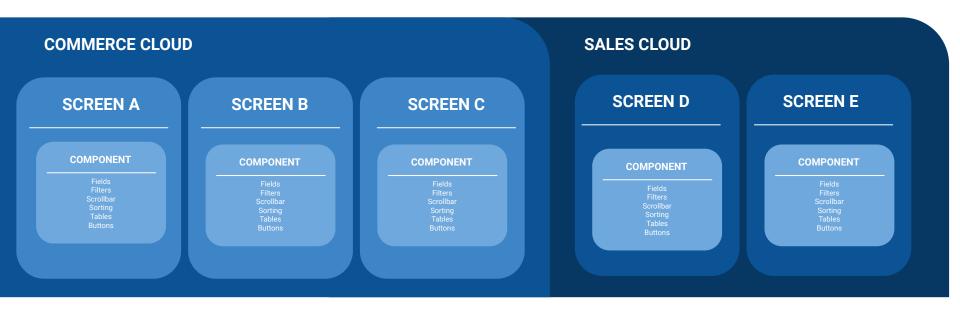
	Туре	Who	When	Requirements	Characteristics
1	Performance Testing (out of scope CRT)	Developers Testers	Dev Int UAT	Non-Functional	<ul> <li>Did we build performing software?)</li> <li>Performed after functional testing</li> <li>Measures response times under different load conditions</li> <li>Runs as part of or after the software build process</li> <li>Test Components and underlying Infrastructure</li> </ul>
2	Usability Testing (out of scope CRT)	Testers UX Designers	Dev Int UAT	Non-Functional	<ul> <li>Did we build usable components?)</li> <li>Can start early for having productive feedback loops in the design phase</li> <li>Accessibility testing (for deaf and blind people)</li> <li>Executed at application level (post build and deployment)</li> </ul>
3	Security Testing (out of scope CRT)	Developers Testers	Dev Int UAT	Non-Functional	<ul> <li>Verification activity (did we build secure components?)</li> <li>Penetration tests, static code analysis, vulnerability scanning, etc.</li> </ul>



# **Screen Components**



# **What are Component Tests?**



#### **PURPOSE**

Small Test
Test Functional Requirements Thoroughly

#### **POSITIVE & NEGATIVE TESTING**

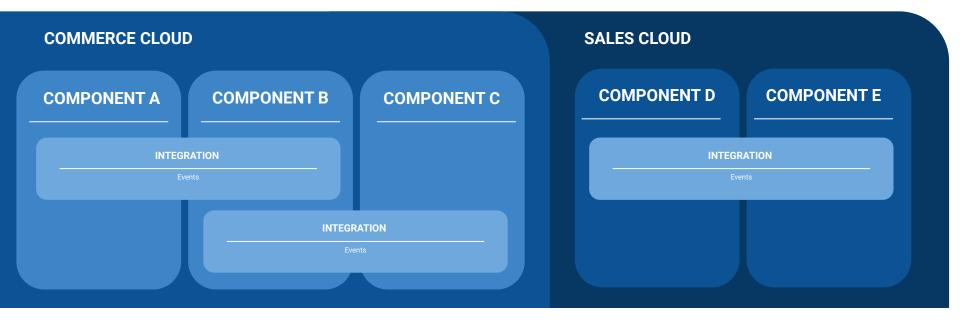
Allowed Digits
Allowed Character Formats
Active / Not Active
Visible / Not Visible
Mandatory / Optional

#### **DATA PRECONDITIONS**

UI Automation API Database Query Stubs / Mocks



# What are Integration Tests?



**PURPOSE** 

**POSITIVE & NEGATIVE TESTING** 

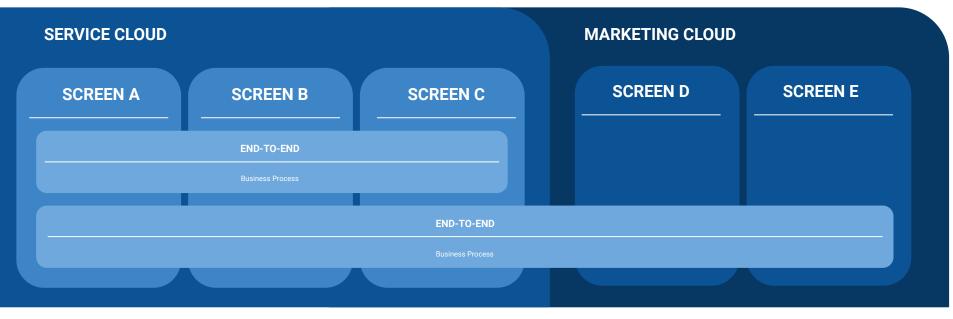
**DATA PRECONDITIONS** 

Small Tests
Test Functional Requirements Thoroughly

Navigations Data Correctness UI Automation API Database Query Stubs / Mocks



# **Acceptance Testing using End-to-End tests**



**PURPOSE** 

**REMARKS** 

**DATA PRECONDITIONS** 

Larger Tests testing Real World User Behavior Test Business Requirements / Processes Design End-to-End Test Cases More suitable for Happy Flows Suitable to be driven by data (data-driven testing) You cannot finish the flow? Automate intermediary steps

UI Automation (start to finish)



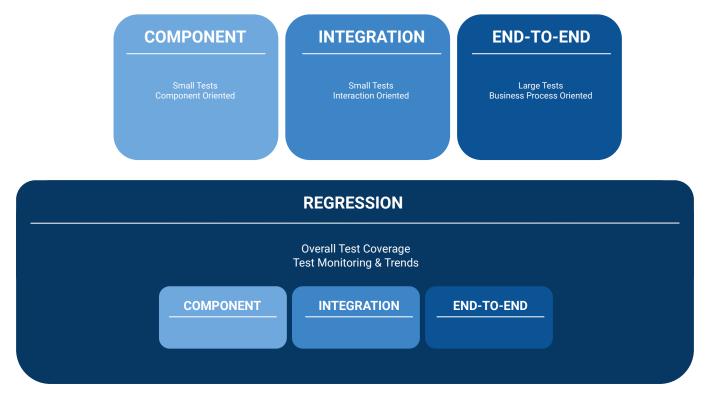
# Three types of Use Case Flows you can use for creating Tests

# ALTERNATE FLOWS COMPONENT Any INTEGRATION Any ERROR / EXCEPTION FLOWS COMPONENT Any INTEGRATION Any



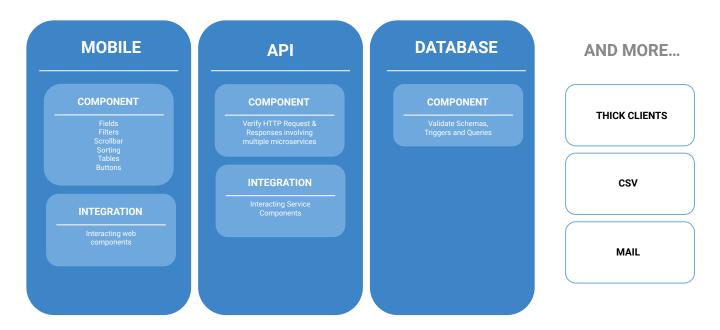


# How do you build a regression test suite?

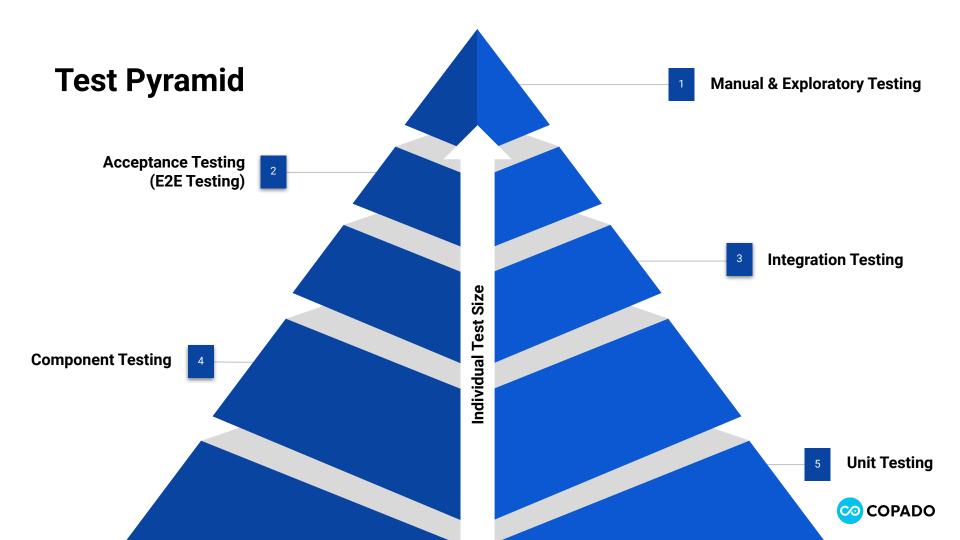




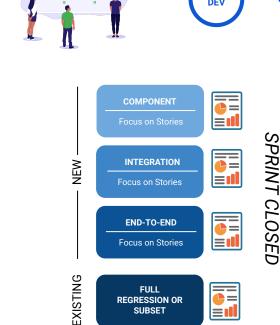
# **Supporting different Technologies & Protocols**







# **In-Sprint Automation & Continuous Testing**







#### **FULL REGRESSION**

Focus on completed stories, features and application releases Longer Test Executions (try not exceed 1 hour runs, if so, add extra robots and run in parallel)

## On Demand

PROD

#### **Scheduled**

- Nightly / Daily
- Hourly

## **Event Triggered**

- Commit
- Promotion
- Promotion Deployment





# **Test Automation**

**Principles** 

# **Test Automation Principles**

Automation requires a robotic perspective!







# **Copado Robotic Testing**

**Fundamentals** 

# **Copado Robotic Testing Platform**

## **Featuring**

- User-friendly keyword driven automation built on Robot Framework
- Pre-built Libraries Out-of-the-Box (QWeb, QForce, QVision and more..)
- Online Test Development Environment
- Centralized Execution & Reporting Environment
- Test infrastructure setup and maintained in the cloud for you
- Option to Customize and Extend Automation Capabilities
- Pre-built integrations with e.g. Azure, Jira, BrowserStack, Slack, Teams.
- Support for On-Premise
- Copado Success Community for Support



# **Copado Robotic Testing Platform License Options**

How much testing do you anticipate?

How many users and parallel robots will you need?

Do you need additional users or robots?

## **Standard**

Entry level option for those who are just starting in test automation

- √ 5 users
- ✓ 1 parallel robot

## **Professional**

Intermediate option for those who want to extend test automation scope beyond the basics

- ✓ 10 users
- ✓ 3 parallel robots

## **Enterprise**

For those who want to automate across multiple systems and end-to-end business processes

- ✓ 25 users
- ✓ 8 parallel robots

## ELA

For teams that need the highest security and availability

 Customized package of users and robots

### **Additional Users**

**Additional Robots** 

On-Premise Robots **Additional Users** 

**Additional Robots** 

On-Premise Robots

**Additional Users** 

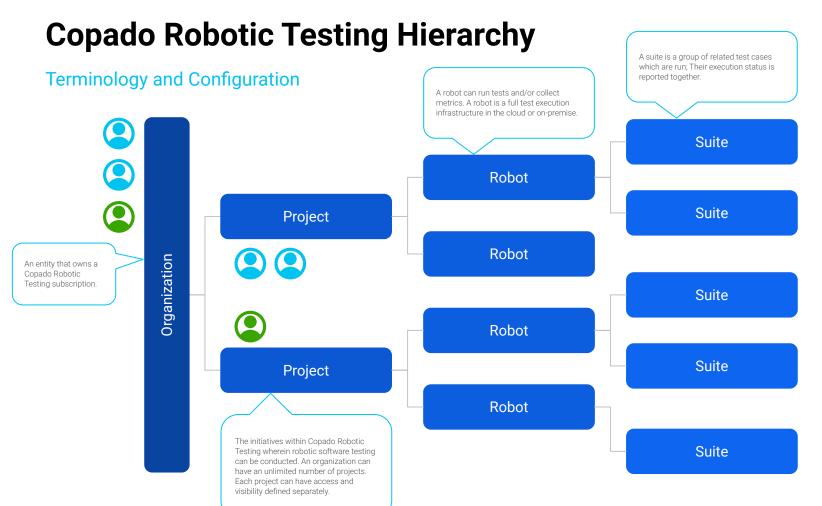
**Additional Robots** 

On-Premise Robots

**Customer Success** 

% Of Total Contract Value











# **Project Structure**

Setup

# Which Projects do I need to configure?

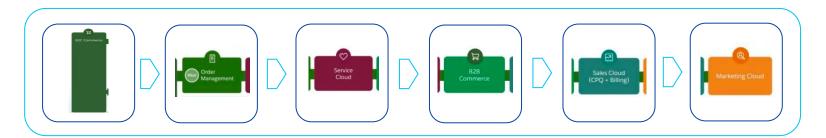


## **Scope of Testing: Single Application**

- B2C Commerce
- Order Management
- Service Cloud
- B2B Commerce
- Sales Cloud (CPQ + Billing)
- Marketing Cloud

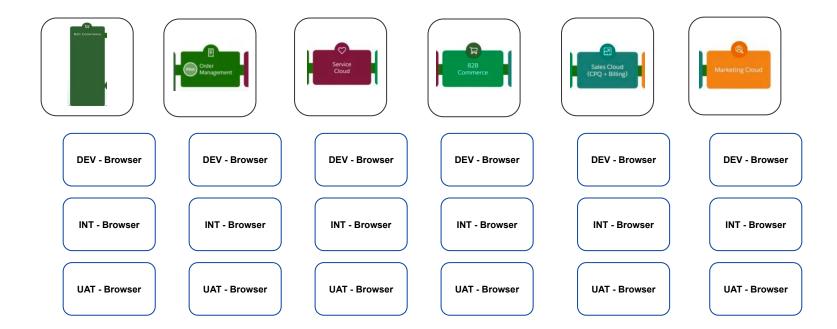
## **Scope of Testing: Multiple Applications**

Cross Application (end to end, user journeys, chain)



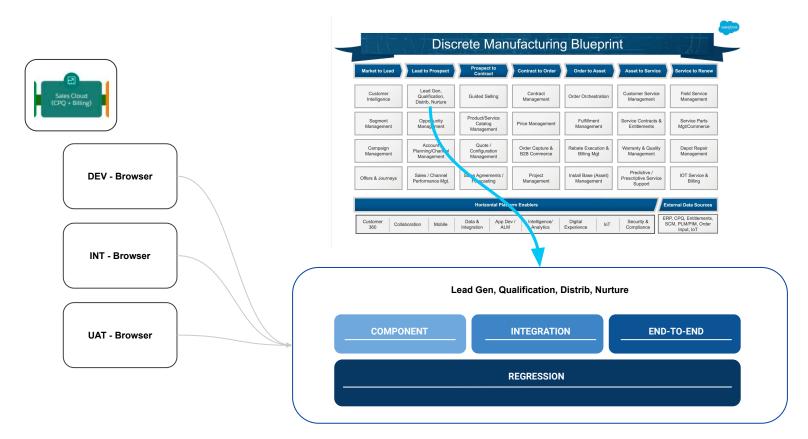


# Which Robots do I need to configure?



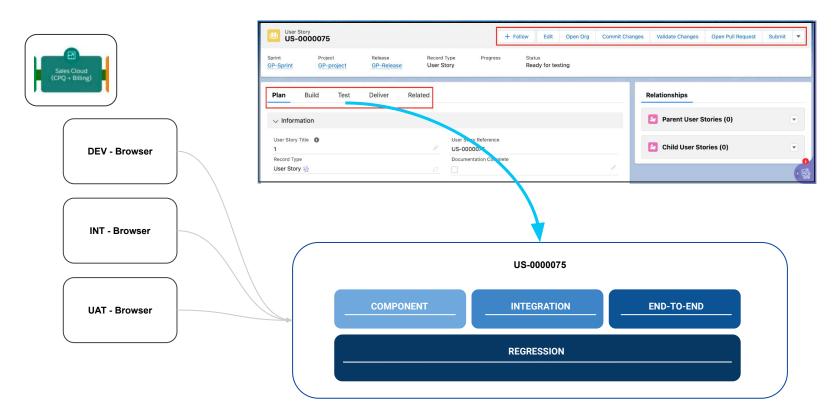


# Which Suites do I need to configure?





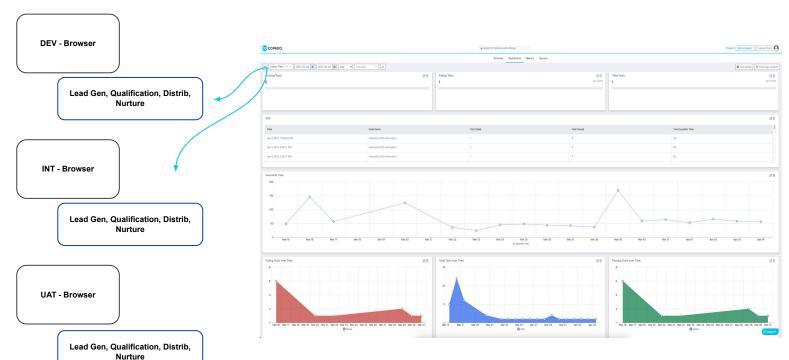
# Which temporary Suites can I configure?





# Filter Project Dashboard by Robots and Suites

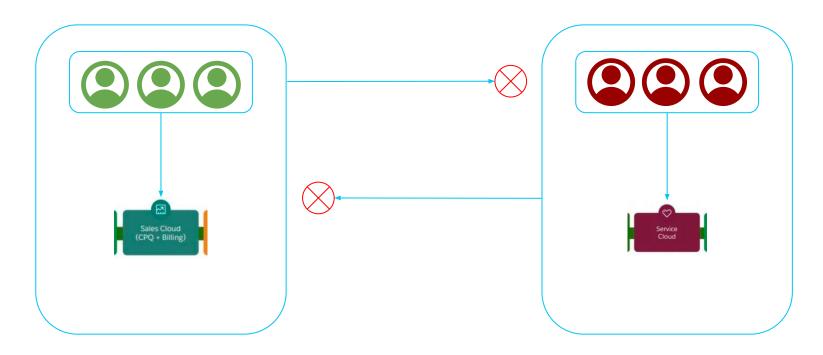
Collect data over time for all your environments and value streams







# **Security Considerations for Projects**







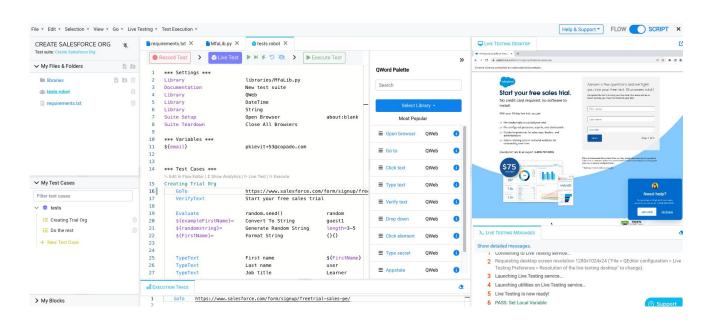
# **QEditor Basics**

Online Test Development

# **QEditor** Online Test Development Environment

## For every Suite

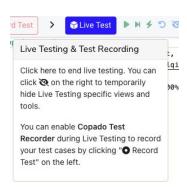




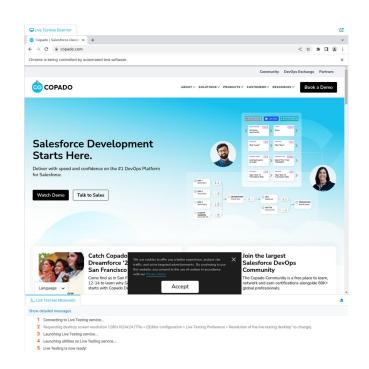


# **Live Testing**

## Watch your tests run! Run complete, single step or selection



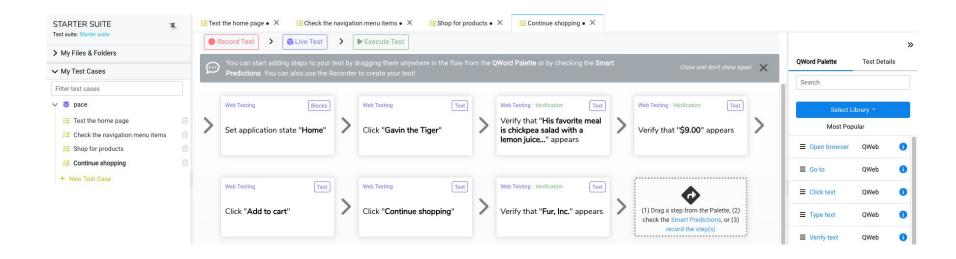






## The Flow Editor

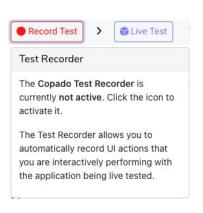
## User Interface

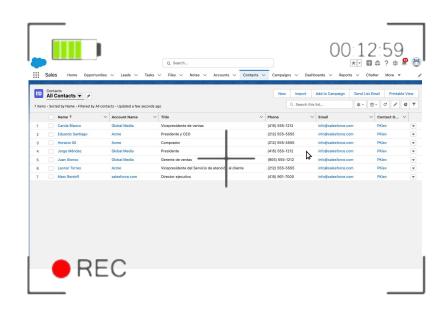




# **Test Recorder using Live Testing**

## Records user input and generates keywords







#### As a user I can view all contacts

LaunchApp ClickText ClickText Sales Contacts All Contacts



