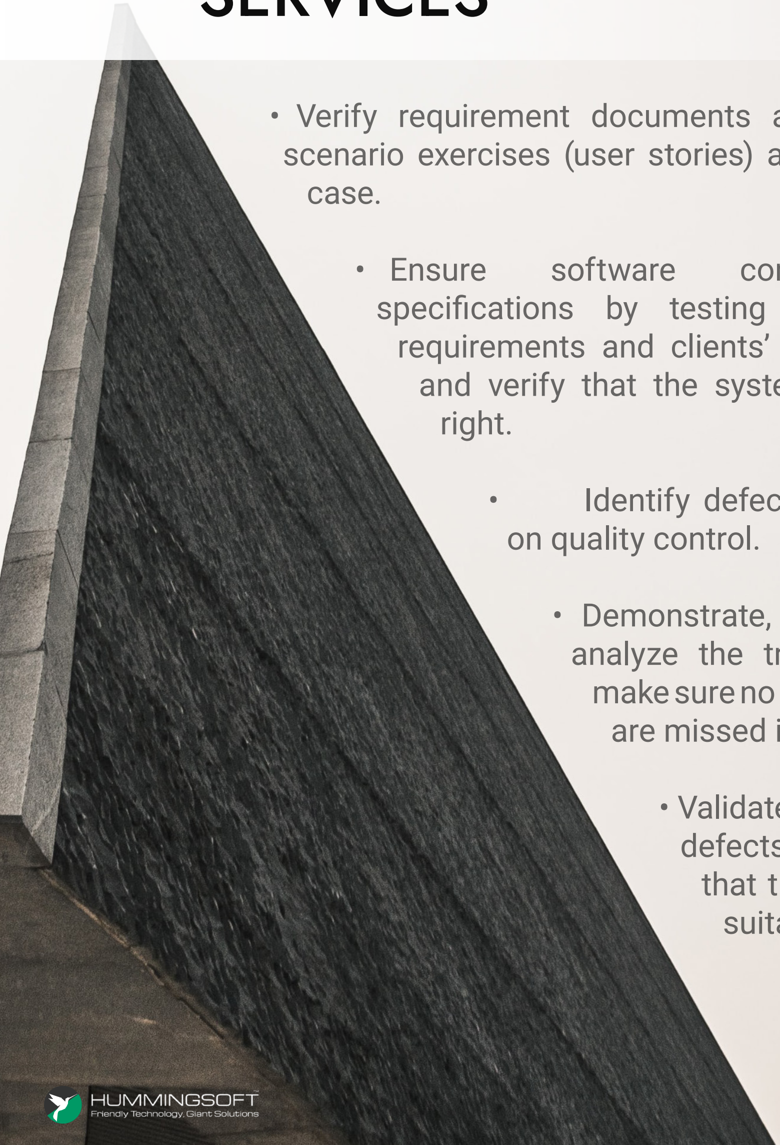
3RD PARTY INDEPENDENT AGILE SOFTWARE TESTING SERVICE AUTONOMOUS UAT

HUMMINGSOFT™
Friendly Technology, Giant Solutions

Introduction to:

AGILE SOFTWARE TESTING SERVICES



- Verify requirement documents and practical scenario exercises (user stories) and build test case.
- Ensure software conforms to specifications by testing against its requirements and clients' expectations and verify that the system was built right.
- Identify defects and focus on quality control.
- Demonstrate, inspect, and analyze the traceability to make sure no requirements are missed in the build.
- Validate the rectified defects and certify that the product is suitable for use.

ABOUT US

A software testing and quality assurance (QA) automation vendor with 9 years of experience and 14 successfully completed software testing projects, Hummingsoft is recognised under the Malaysian Software Testing Board (MSTB) under the Malaysian Administrative Modernisation and Management Planning Unit (MAMPU) as a cluster company. We apply our innovative and progressive approaches to eliminate errors and bugs, speed up release cycles, and maximise performance.

TRACK RECORD

Project Title	Year	Organization / Agency
Iwant2b	2013	PEMANDU
Labour Market Management System	2013	Institute Of Labour Market Information & Analysis
MRT Project Executive Dashboard	2014	MRT Corp, MOT
Data Analytics on Budgeting System	2015	Petronas Berhad
Data Analytics Training	2016	Rehanstat Sdn Bhd
Bakri Plantation System	2016	Bakri Sdn Bhd
Education Data Repository 1	2017	Kementerian Pelajaran Malaysia
Plantation Information Dashboard	2019	Pinfosys Sdn Bhd
Grant Management System	2019	Prime Minister’s Office
ICP Management System	2020	Technology Depository Agency Bhd -TDA, MOF
Education Data Repository	2021	Kementerian Pelajaran Malaysia
Project Management Training	2021	Mimos Berhad
Humworks Project Management	2021	MKRS Bumi (M) Sdn Bhd
Education Data Repository 2	2021	Kementerian Pelajaran Malaysia

QA Automation Services

We expedite clients’ QA testing with training, test automation planning, implementation and maintenance of test cases for a seamless integration CI/CD and RPA Automation Testing technology

HUMMINGSOFT
Friendly Technology, Giant Solutions

TESTING TEAM ORGANIZATION CHART

TEST MANAGER

- Create test strategy documents
- Make sure test tasks are scheduled into the correct sprints
- Suggest and implement ideas for efficiency
- Negotiate for resources and time
- Provide reports for test coverage and defects
- Keep a close eye on estimates vs actuals

TEST LEAD

- Detect management and trend spotting
- Help to prioritize the backlog based on risk impact and dependencies
- Identify and reporting scope creep
- Analyze requirements for conflicts and inconsistencies
- Help to create acceptance criteria
- Ask questions and prompting discussions
- Carry out demonstrations and overviews for sign off and quality checks
- Liaise with the UAT team
- Focus on preventing bugs rather than finding them

TESTERS

- Design automated tests for regression
- Prepare test data efficiently
- Support build and release management

HUMMINGSOFT
Friendly Technology, Giant Solutions

AGILE SOFTWARE TESTING TIMELINE

REQUIREMENT PHASE

8 Days	Requirement Static Test
14 Days	Test Case Development
2 Days	Confirmation of Sprint / Release Backlog

UPON COMPLETION OF EACH SPRINT

3 Days	Regression Test (previous sprint / release)
5 Days	Integration Test (sprint integration)
4 Days	System Test (by release)

HUMMINGSOFT
Friendly Technology, Giant Solutions

TYPES OF SOFTWARE TESTING

PERFORMED BY US

Early Testing (Static Testing)

To assess the tracibility of user requirement, the testability of the software and to detect defects before going into implementation

Compability Testing

To determine whether the software is compatible with other software operating in the same environment

Conversion Testing

To determine whether data that is being transferred from existing software to new software is done

Functional Testing

To determine whether all functions of a build are producing the required output as specified in the functional requirements/ specifications

Portability Testing

To determine whether the software functions correctly when it is transferred from one environment to another

Acceptance Testing

To determine whether a system satisfies its acceptance criteria and to enable the customer to determine whether to accept the system

Integration Testing

To determine whether exposure of defects in the interaction between build when they are integrated

System Testing

To determine End-to-End testing of "integrated" hardware and software components to see if inputs deliver expected outputs

Regression Testing

To determine whether additional bugs have not been introduced, from the modified build

SOFTWARE TESTING SCOPE OF WORK

Early Review

Business Processes and Requirements Evaluation

• Evaluate the current user requirement specification for correctness, consistency, completeness, accuracy and readability.

• Evaluate the to-be business processes for correctness, consistency, completeness, accuracy and readability.

• Verify the Project has plans in place to transition from the current business processes to the to-be business processes.

• Build test plans, test cases, and procedures that conform to the Project- defined test document purpose, format and content.

Software Testing

Software Integration Testing

• Validate the integration test designs, test cases, and procedures are complete, traceable to requirements, and consistent with the test plan.

• Evaluate the test environment, tools, and procedures used for integration testing of system modules.

• Validate the integration test plan is traceable to the requirements

• Perform Integration test and verify the test results are verified are correct and all build functions for the specified Sprint / Release has been tested, and that the tests are appropriately documented, including formal logging of errors found in testing.

• Use the integration test results to verify the system satisfies the test acceptance criteria.

System Testing

System Testing

- Validate the system test designs, test cases, and procedures are complete, traceable to requirements, and consistent with the test plan. Evaluate the test environment, tools, and procedures used for system testing.

- Validate the system test plan is traceable to the system requirements.

- Verify that a sufficient number and type of case scenarios are used to ensure comprehensive but manageable testing and that tests are run in a realistic, real-time environment.

- Perform System Test and verify that all test scripts are tested, with step-by-step procedures, required pre-existing events or triggers, and expected results.

- Use the system test results to verify the system satisfies the test acceptance criteria.

Acceptance Testing

Acceptance Testing

- Acceptance procedures and acceptance criteria for each build defined, reviewed, and approved by the customers prior to test and the results of the test must be documented. Acceptance procedures must also address the process by which any software product that does not pass acceptance testing will be corrected.

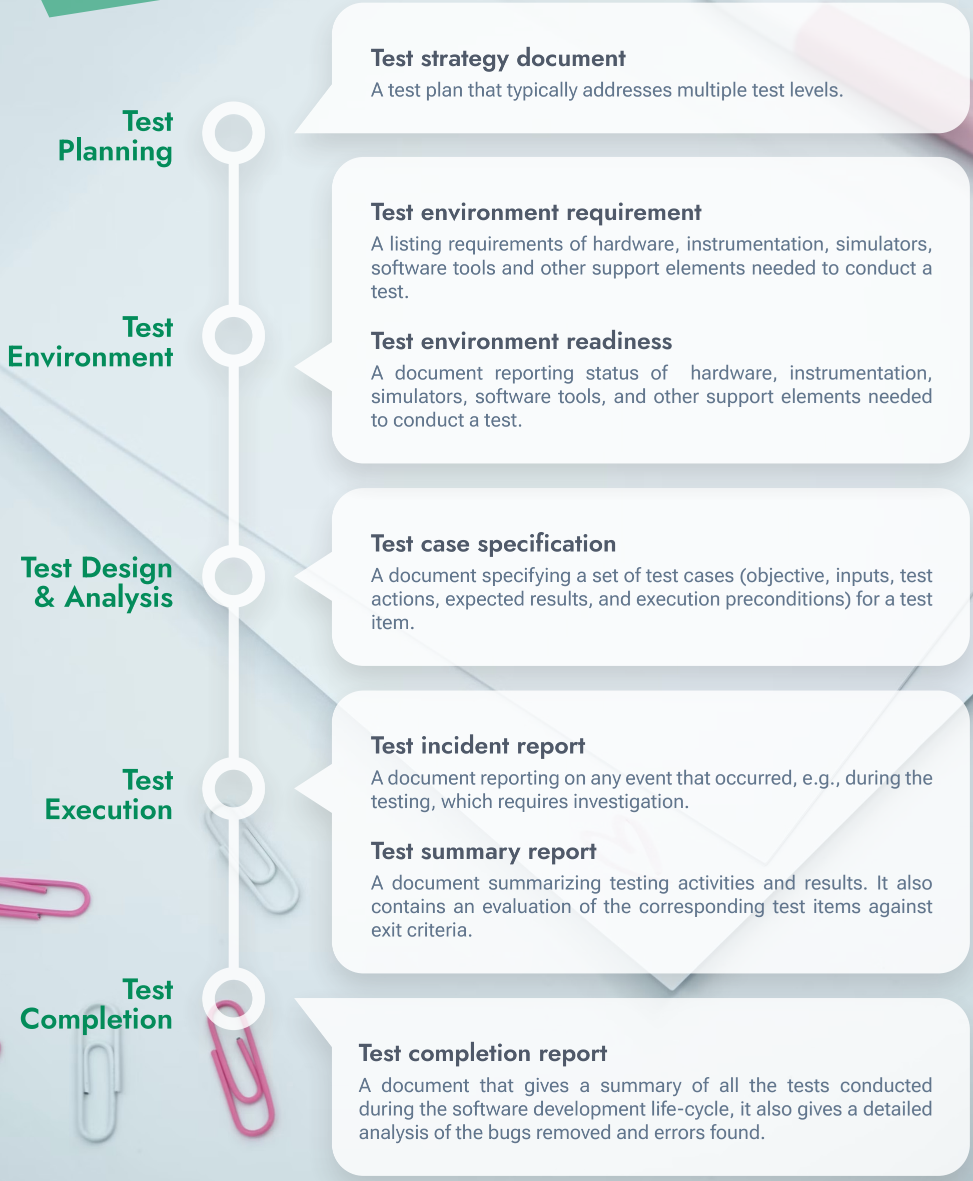
- Verify the acceptance test plans, test cases, and procedures

- Verify the acceptance test plan addresses test coverage of acceptance requirements, expected results, and the feasibility of operation and maintenance.

- Prepare the users to perform the Acceptance Testing.

- Use the system acceptance test results to verify the system satisfies the test acceptance criteria.

DOCUMENTS & REPORTING PROVIDED



Test Planning

Test strategy document

A test plan that typically addresses multiple test levels.

Test Environment

Test environment requirement

A listing requirements of hardware, instrumentation, simulators, software tools and other support elements needed to conduct a test.

Test environment readiness

A document reporting status of hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test.

Test Design & Analysis

Test case specification

A document specifying a set of test cases (objective, inputs, test actions, expected results, and execution preconditions) for a test item.

Test Execution

Test incident report

A document reporting on any event that occurred, e.g., during the testing, which requires investigation.

Test summary report

A document summarizing testing activities and results. It also contains an evaluation of the corresponding test items against exit criteria.

Test Completion

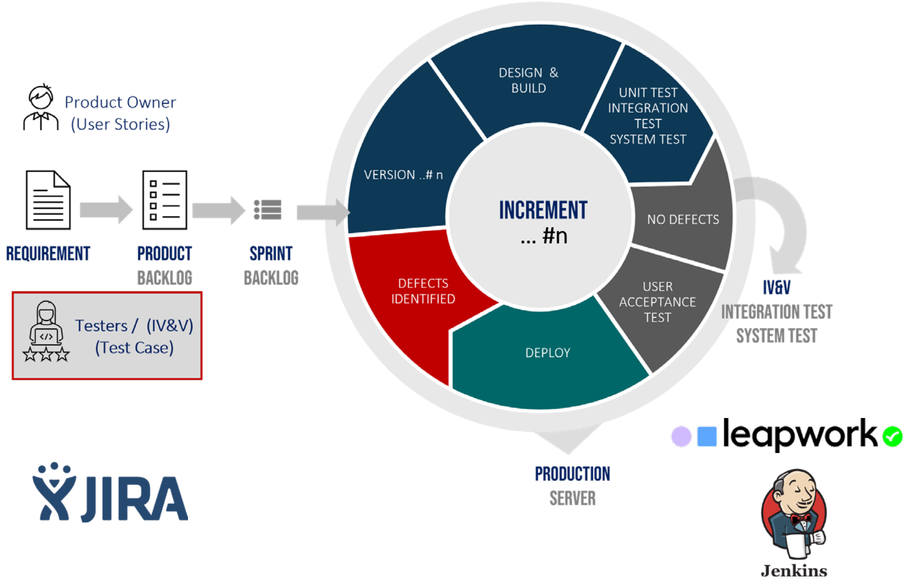
Test completion report

A document that gives a summary of all the tests conducted during the software development life-cycle, it also gives a detailed analysis of the bugs removed and errors found.

IV&V AGILE SOFTWARE TESTING SERVICES

Requirement Review Stage

As an Independent Verification and Validation Software Tester, our services start in tandem with both the end user and the development team to clarify requirement assumptions and make them explicit, and to explain any ambiguities in those stories before developers start coding. By combining responsibilities, testers, business analysts and developers each do what they do best—turn user requirements into high-quality applications. When you user stories are created, the acceptance criteria is defined. This drives testing and validation of the user stories.



Agile Test Strategy

Testing in traditional development typically includes a test plan. But a well-documented test plan is not common in Agile development. Instead, Agile testers need to be flexible and ready to respond to shifts in requirements. So, there needs to be an Agile test strategy rather than an Agile test plan.

Our Agile Test Strategy includes purpose (defined by the user story), objectives (test cases), scope (what needs to be tested) and Methods (how tests will be run).

Jira will be used to perform this activity.

IV&V AGILE SOFTWARE TESTING SERVICES

Upon Completion of Internal Testing Stage

Continuous testing is the only way to ensure continuous progress. Agile Testing provides feedback on an ongoing basis and till final product meets the business demands. The entire team including analysts, developers, development team testers and IV&V Testers test the application and after every release, even the customer performs the User Acceptance Testing.

The client's engagement in performing the User Acceptance Testing at the end of every release will enable them understand the product better and propose changes early. Continuous feedback shortens the response time and incurs lesser cost.

Defects captured during the IV&V Regression, Integration and System Testing are fixed as they are raised within the same Sprint/Release. Customer changes recorded during the Acceptance Test are developed in the subsequent Sprint/Release. This ensures clean code at any milestone of development.

Our IV&V Software Testing Service documentation is lightweight instead of comprehensive documentations like the waterfall methodology. Autonomous tools like Jira and Jenkins are used to build reusable test cases and move our focus on the essence of the test rather than the incidental details. Test Cases are written along with the requirements. This approach is called Acceptance Test Driven Development (ATDD). This contrasts with testing as a last phase in Waterfall Testing.

Acceptance Test Driven Development

In the Acceptance Test Driven Development (ATDD) method, the code is developed based on the test-first approach directed by Acceptance Test Cases. The focus is on the acceptance criteria and the Acceptance Test Cases written by the IV&V testers during User Story Creation in collaboration with the customer, end users and relevant stakeholders.

step 1 Write Acceptance Test Cases along with user stories in collaboration with the customer and users.

step 2 Define the associated acceptance criteria.

step 3 Develop code based on the acceptance tests and acceptance criteria.

step 4 Run the acceptance tests to ensure the code is running as expected.

step 5 Automate the acceptance tests, Repeat Step 3 – Step 5 until all the user stories in the iteration are implemented.

step 6 Automate the regression tests.

step 7 Run the automated Regression Tests to ensure Continuous Regression.

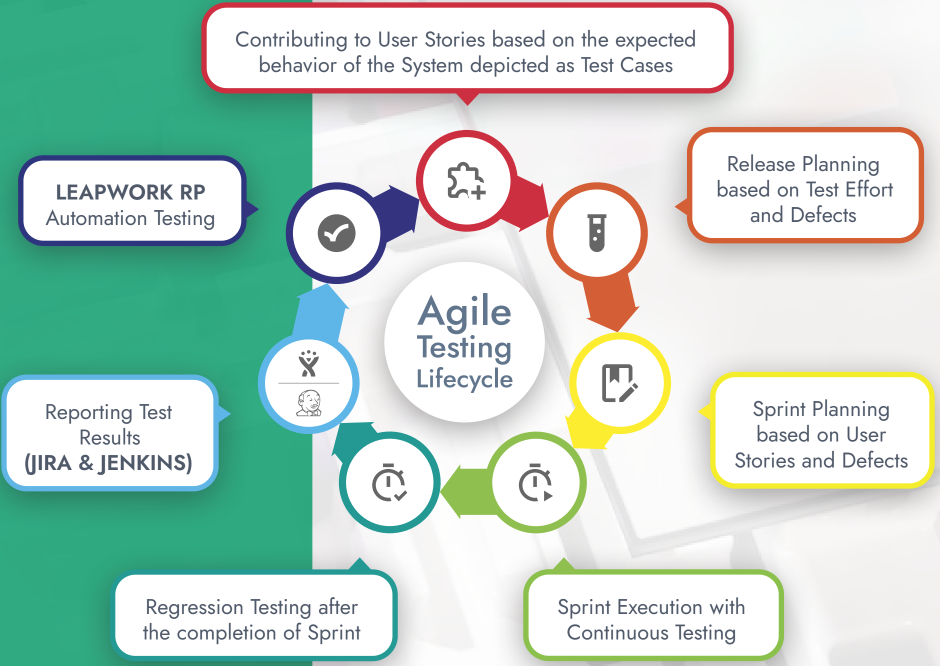
Product Quality Reporting

Features embedded in Jira and Jenkins will be used for automating and reporting of product quality metrics. This helps in maintaining transparency with both the client and the development team. It also reduces delays in reporting and communicating defects to the developers. All relevant quality metrics are delivered as soon as the Automation Test Blocks are executed.

Product Quality Metrics:

- Test pass/fail
- Defects found/fixes
- Test coverage
- Test pass/fail rates
- Defect discovery rates
- Defect density

“To secure overall product quality, the Agile team needs to obtain customer feedback on whether the product meets customer expectations. This needs to be carried out at the end of each release, and the feedback will be an input for subsequent releases.



Agile Testing Lifecycle

Contributing to User Stories based on the expected behavior of the System depicted as Test Cases

Release Planning based on Test Effort and Defects

Sprint Planning based on User Stories and Defects

Sprint Execution with Continuous Testing

Regression Testing after the completion of Sprint

Reporting Test Results (JIRA & JENKINS)

LEAPWORK RP Automation Testing

leapwork

ENTERPRISE SOFTWARE TEST AUTOMATION TECHNOLOGY

Enterprise test automation and RPA that's accessible to *more people*

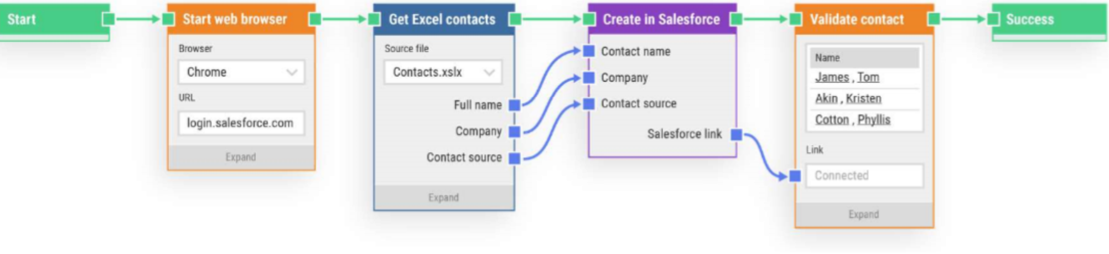
Leapwork's visual, no-code automation platform makes it easy for business and IT users to automate repetitive processes, so enterprises can adopt and scale automation faster.

A universal, visual automation language that everyone can understand

From business analysts and QA engineers to corporate managers, there are people across your organization who know what software tests and business processes to automate, yet they don't necessarily have the technical expertise traditionally required to automate them.

That's where Leapwork comes in.

With Leapwork, the pace of success is no longer determined by the availability of IT or development resources. Business users and QA teams can now build automation with the universally recognized language of flowcharts rather than code. Simply show Leapwork what to automate and watch the platform express the process in a visual way, with intelligent building blocks that indicate every step in a process and handle complex logic and data transformations in the background.



> “The pace you can automate is astounding and blows the competition out of the water”

Lawrence Williams,
Head of Quality Assurance, Telrock

Why Leapwork?

97% productivity gains
compared to manual workflows

Unlock innovation by eliminating repetitive, time-consuming tasks and facilitate agile teams that can focus on value creation.

90% reduction
in critical incidents

Rapidly reduce the risk of costly downtime or revenue loss associated with errors, and delight customers with consistent quality.

Reduce the cost of hiring
and upskilling

Business analysts and QA teams can leverage existing skills to automate tests and processes, reducing the need to hire or upskill more people.

3-10x faster time to market
regularly achieved by customers

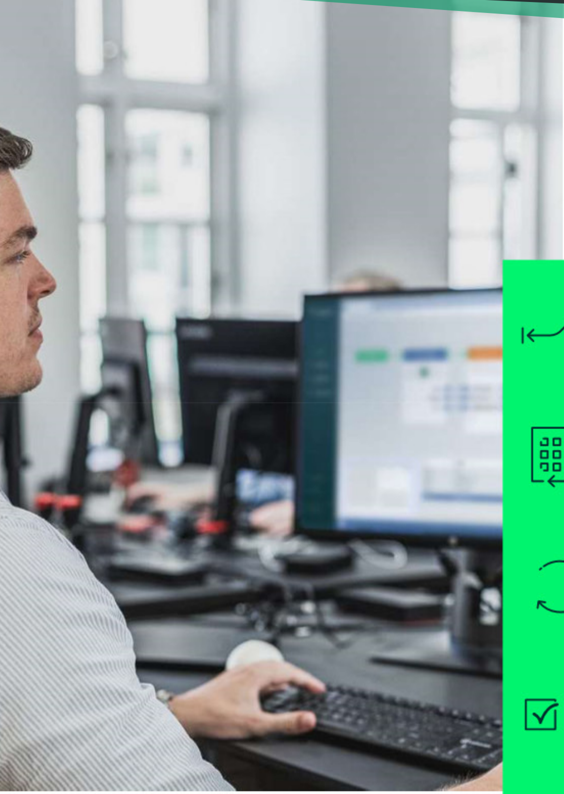
Sharpen your competitive edge, ensure a high degree of responsiveness to customer needs and outpace changes in the marketplace.

Less than 10 minutes
to automate a test or process

With the shortest learning curve on the market, enterprises can benefit from automation and scale faster.

Minimize maintenance
and solve problems quickly

With AI and built-in reusability, Leapwork makes it easy to detect and solve problems faster, for more agile teams with fewer maintenance bottlenecks.



Examples of automation with Leapwork

- End-to-end testing across complex IT environments
- Regression and functional UI testing
- Continuous testing with plugins to existing CI/CD pipelines
- Start-of-day checks for business-critical processes
- Validate SaaS and packaged application updates and customizations
- Robotics Process Automation (RPA) serving business domains from Finance and Legal to HR and Sales

> "The ROI is a no-brainer"
>
> David Pound
>
> Head of Lending Technology, Investec

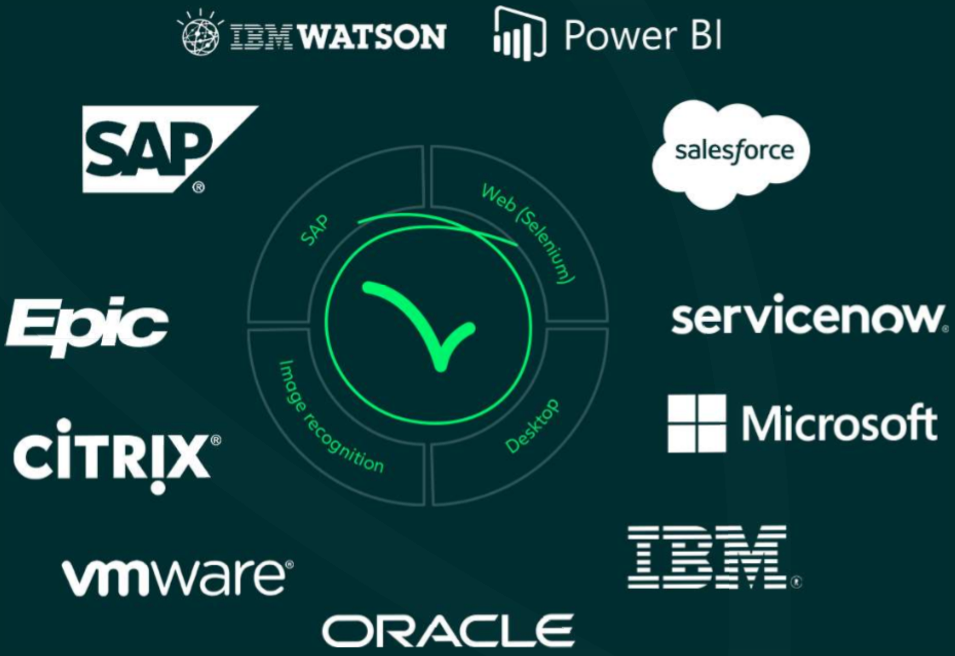
Remove the maintenance burden and scale automation easily

To facilitate agile delivery pipelines and sustainable, scalable automation, Leapwork comes with several capabilities that minimize the maintenance workload. Reusability is built in, so automation cases and common components can be created and updated in one place then shared across teams and suites.

If something breaks, the platform takes you to the root cause and provides a hyper-visual representation of the issue in three ways: a video recording of the run, a debug version of the automation flow, and an activity log. All three are correlated to give immediate visibility of the problem, and every run includes detailed data-level insights that make it easy to inspect outcomes.

A uniquely *simple* way to automate across complex IT environments

Leapwork might be simple to use, but it runs on a powerful technology stack that enables the platform to work seamlessly across your organization's application portfolio. Use the same visual approach to automate anything from desktop-based systems, Mainframes and ERP software that serve your back-office to web platforms, mobile web apps, and virtual desktops such as Citrix.



Robust, lightning-fast SAP automation

Leapwork comes with dedicated SAP capabilities to provide a powerful yet intuitive way to automate SAP that delivers lightning-fast execution. Business users, application specialists, and QA teams can leverage their existing expertise to automate SAP with Leapwork's dedicated SAP recorder. Perform any step in SAP GUI, validate values along the way, then let Leapwork automatically generate the automation flow.

> “We've achieved a level of coverage that's not possible with alternative solutions on the market”

Claes-Jan Wolff

Senior IT Architect, GF Forsikring

Comes with a range of tools to facilitate rapid enterprise-wide adoption and faster results

Reusable cases and components

Don't build functionality twice. Turn automation components into reusable assets to share across projects and teams, for faster automation design and efficient maintenance.

Built for DevOps

REST API plugins to all popular CI/CD tools and orchestrators such as Jenkins, Azure DevOps, and many more. Trigger automation to run on-demand, and create custom integrations to on-premise or cloud-based systems.

Drive automation with data and parameterize

Run cases with input from spreadsheets, databases, and web services. With the REST API, make calls to HTTP endpoints to fetch data. Easily map data, define variables, and pull into reusable sub-flows. Parameterize for accuracy, efficiency and scalability.

Workflow Management

Get enhanced automation governance using built-in workflow management. Track progress with versioning, segregate duties, assign automation cases to the right contributors and handover flows easily.

Teams Management

Supports strict access control schemes and Active Directory infrastructure. Enable contributors to work independently on protected assets or share automation cases and components easily across projects.

Dashboards and reporting

Visualize and understand vast amounts of data with integrations to Microsoft Power BI and Tableau. Leapwork also comes with built-in performance reports and video-based reporting.

Advanced Scheduling

Define when automation should run: at pre-determined times, after specified intervals, or trigger on-demand as part of the automation pipeline. Run across multiple devices, OS and browser combinations, or on virtual and cloud-based machines such as Amazon EC2 and Microsoft Azure. A calendar assistant gives an instant overview of available resources.

Governance and security

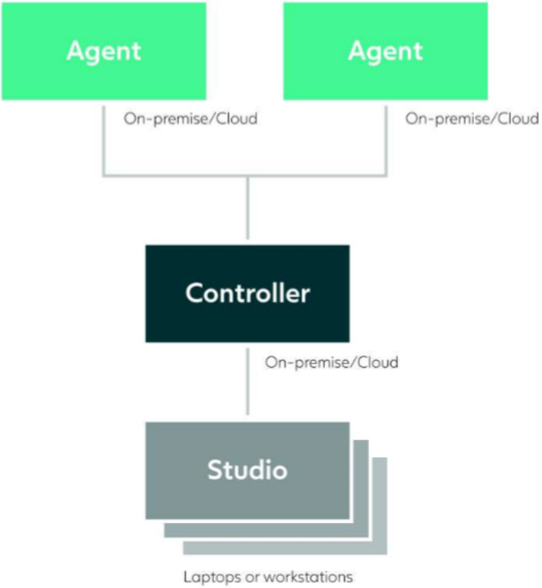
All Leapwork traffic is encrypted (e.g. TLS 1.2 compliant), including internally between platform components and at rest using SHA-256 encryption. Keep all historical runtime data, manage roles and permissions, and deploy securely into existing infrastructure, whether on-premise, in hybrid architectures, or in the cloud.

Extended compliance tools

Designed for enterprises with extended regulatory requirements or internal compliance standards: get tamper-proof audit logs and absolute version control to verify and guarantee that automation approved in a specific context remains unchanged.

Deploy seamlessly to your needs

Leapwork consists of three components: Studio (where automation design takes place), Controller (the server) and Agent (the runner). The diagram shows an example configuration. This enables multiple team members to work together immediately to design and preview automation cases on their own laptops or workstations (PCs). Agents are then shared for scheduled execution of the automation cases e.g., in separate test and pre-production environments, when they are finished.



World class support

Leapwork provides a range of support options aimed at delivering rapid and effective adoption, fast return on investment, and easy scaling as automation requirements increase.

Support includes

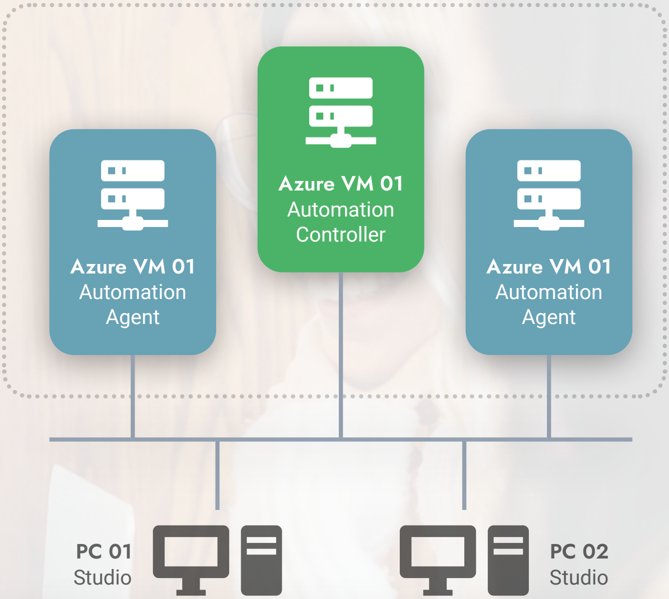
- On-demand technical chat with automation specialists
- A dedicated Customer Success Manager
- Quarterly Business Reviews to support a KPI-driven approach
- Access to the Customer Portal for easy collaboration
- Access to the Leapwork learning center and library of video tutorials

> "Leapwork's technical support is the best I've ever experienced. **Their experts really are experts.**"

Prachi Yakar

Automation Specialist, JF Hillebrand

LEAPWORK PLATFORM
CONFIGURATION & SPECIFICATION



Specifications

Azure VM 01, 02 & 03 (1 x Controller, 2 x Agent)

- Windows Server Service Pack 1
  (2008 R2, 2012, 2012 R2, 2016, 2019)
  - 16GB RAM (min)
  - 4-6 CPU Cores (min)
  - 60GB (SSD, SAN or similar)
- Needs to be configured to accommodate the workload
  - Number of automation flows
  - Complexity and duration of automation flows
  - Frequency of execution of automation flow

Studio PC 01 & Studio PC 02

- Installed on the user's workstation
  (2008 R2, 2012, 2012 R2, 2016, 2019)
  - 8GB RAM minimum (16 GB is recommended)
  - Min Intel i5 CPU at least 2 cores
  - SSD based on internal storage (20GB)

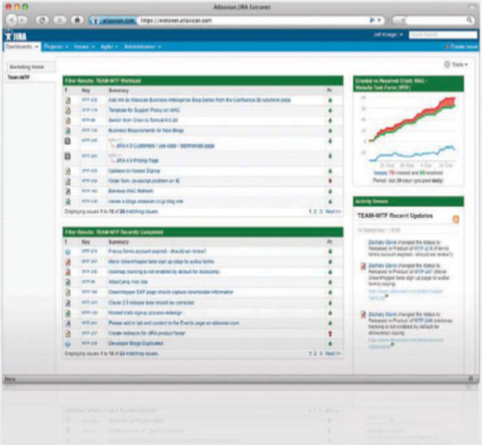
JIRA

TRACK BUGS, STORIES AND EPICS

The centre of your development team

JIRA combines issue tracking, agile project management, customisable workflow, and a pluggable integration framework to increase the velocity of your software development team.

With JIRA at the centre of your development team, delivering quality software on-time has never been easier.



Why JIRA?

JIRA simplifies every step of tracking issues, for everyone involved. Creating, reviewing, and resolving issues is a snap. And JIRA goes much further, supporting complex project management and agile development processes.

Who uses JIRA?

JIRA is used by more than 11,500 organisations, located in over 107 countries. Customers include: Adobe, Cisco, HP, Intuit, Oracle, Samsung, Yahoo!, Citigroup, Credit Suisse, Boeing, BP, McDonalds, Shell, Sony, Disney, Verizon, CERN and NASA.

Many open source projects rely on JIRA including Apache, Codehaus, Flex, Hibernate, JBoss, MuleSource, Sonatype, Spring, Zend and more.

> I don't know how we got by without JIRA. A wonderful tool. You can forget Bugzilla. JIRA beats it hands down.
>
> **—Jason McKerr of NACSE**

Bug and Issue Tracking

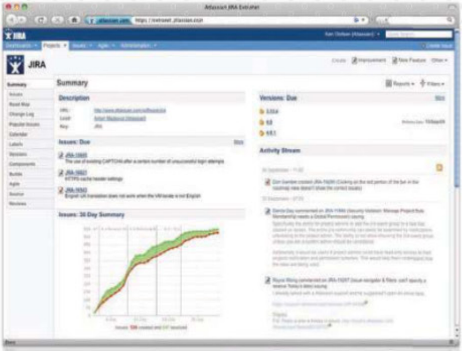
Simple and flexible defect tracking. Integrated with your source code and development environment to fit the way you work.

Agile Software Development

Whether you are a certified Scrum Master or just starting out with agile methodologies, JIRA meets the needs of your development team.

Project Management

Manage all of your projects in one place. Focus on the task at hand without ever losing sight of the big picture.



JIRA Project Browser

Bug and Issue Tracking

Simple and flexible defect tracking. Integrated with your source code.

Agile Development

Meets your needs whether you are a certified Scrum Master or just starting to go agile.

Project Management

Focus on the task at hand without ever losing sight of the big picture.

Dashboards

Personalise your view using OpenSocial gadgets and share it with the world.

Workflow and Fields

Simple and effective workflow out-of-the-box. Customisable to fit your development process.

Search and Reporting

JIRA Query Language (JQL) brings search and reporting capabilities to a whole new level.

Plugins and Extensions

Customise and add on to JIRA with over 100 plugins and extensions, or build your own.

Agile Project Management Plugin For JIRA

The GreenHopper plugin provides agile project management to simplify both sprint planning and task tracking for your team, leveraging JIRA technology including custom workflows, permissions, OpenSocial gadgets and JQL.

About Atlassian

Atlassian is an innovative global software company with more than 15,600 customers in 138 countries, including 30 of the world's top 50 corporations. Over 400 free plugins have been built and supported by the Atlassian community. Our lustworthy products are designed to stand alone, however, their integration capabilities make them even more useful.

Atlassian Products Include:

JIRA: Issue tracking
Confluence: Enterprise wiki
FishEye: Repository insight
Crucible: Code review
Bamboo: Continuous integration
Clover: Code coverage
Crowd: Single sign-on
JIRA Studio: Hosted development suite

Three ways to try JIRA today!

- Take a guided tour in our JIRA Sandbox
- Try a free 30-day online trial
- Download a fully functional 30-day evaluation

www.atlassian.com/jira/try

Pricing

USERS	PRICE
10 users	Get started for $10 All proceeds go to charity
25 users	$1,200
50 users	$2,200
100 users	$4,000
100+ users	$8,000

System Requirements

Operating System: Any that supports Java
Database: PostgreSQL, MySQL, SQL Server or Oracle
Application Server: Tomcat (included), JBoss, WebLogic or WebSphere
Browser: Internet Explorer, Firefox or Safari

Let developers focus on code

Developers want to focus on code, not update issues. We get it! Open DevOps makes it easier to do both regardless of the tools you use. Now developers can stay focused and the business can stay aligned.

HOW CUSTOMERS ARE BENEFITING

Flo

900%

increase in deployments

50%

decrease in cycle time

Jenkins

CONTINUOUS INTEGRATION AUTOMATION TOOL

Continuous Integration Using Jenkins

HUMMINGSOFT™
Friendly Technology, Giant Solutions

Problem Statement

Traditional way of delivering rapid changes in software development projects consists of:

Intermittent integration

Slow release cycle

Code working only on developer machines

Lot of manual QA testing

'Release Meetings' where everyone takes Go- No-Go decision of the build

Manual Staging and Production deployments

Multiple Integration and operations problems

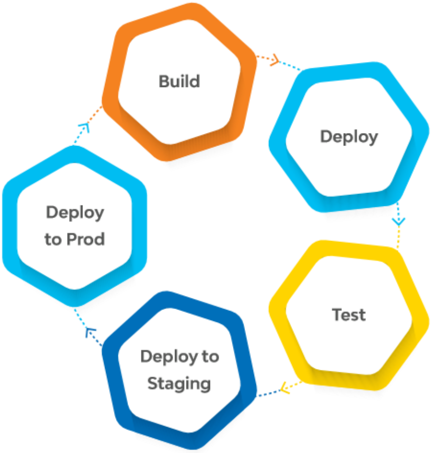
All these results in poor quality of the product, more time to market, more cost, less customer satisfaction, and ultimately adding less value to business.

How to solve this?

Continuous Integration is here to salvage from all these problems. CI is way to increase code quality by continuously integrating the code of different developers on central repository. CI servers trigger build and compilation processes automatically, and notify as soon as something goes wrong. So our tests automatically get executed as and when there is a change in the code. We can have CI server automatically deploying our code to staging and production, if all the tests in given branch are green. This way we can achieve faster build execution, early detection of bugs, increased code quality and faster time-to-market.

Overview of Continuous Integration

"Continuous Integration" or "CI" is a concept, which has been taken as a standard or a parameter nowadays, within our IT services and development. The concept of continuous integration means automating the overall deployment process for an application, after a code has been committed, so as to identify and address the pain points before they become serious issues. So each time a product code is developed, it goes through an automatic CI process. There are several tools for Continuous Integration to help trigger these builds and tests.



Continuous Integration Practices

To make up effective CI and to work above processes smoothly, we have to have some standard practices to be followed from the beginning of our project.

- Maintain a single-source repository
- Make your build self-testing
- Automate the build
- Everyone commits to the mainline every day
- Every commit should build the mainline on an integration machine
- Fix broken builds immediately
- Keep the build fast
- Make it transparent
- Test in a clone of the production environment
- Make it easy for anyone to get the latest executable
- Automate deployment

Advantages of Jenkins

Apart from Jenkins, there are several tools for Continuous Integration, and most of them are open source.

- Cruise Control
- Bamboo
- Buildbot
- Travis CI

GIT: Version Control System

Git is the most widely used modern version control system in the world today, which allows multiple persons to safely work on the same project, without hampering other team members. Git is free and open source designed to handle everything from small to very large projects, with speed and efficiency.

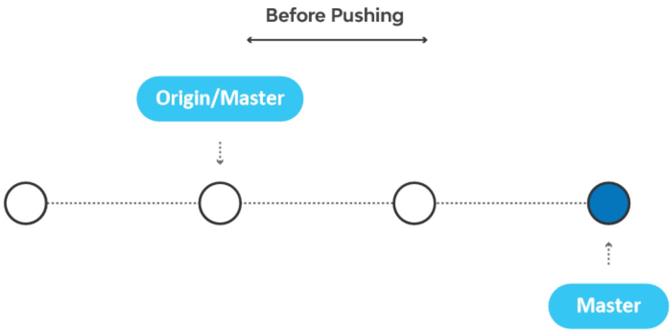
It is distributed and disconnected i.e. every checkout is actually a clone of master repository and every developer has its own repository with its own branches and own commits. We can clone working the copy of a local repository from Git server, which can also add and commit the test scripts that are developed locally and push our changes to the Git.

Git is also used by the Ops team as a version control for automation to store scripts, tools and software configuration.

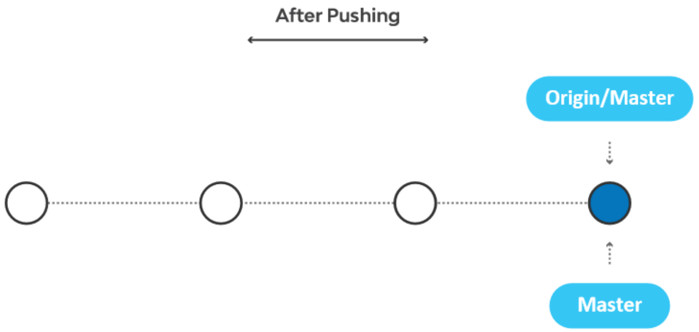
The most important concept of Git is Branching and Merging. Developers can work on their own feature branch without disturbing others work. Branches are very easy to create and merge back. Since these branches are local, these can be used all the time.

Pictorial representation of Git Usage:

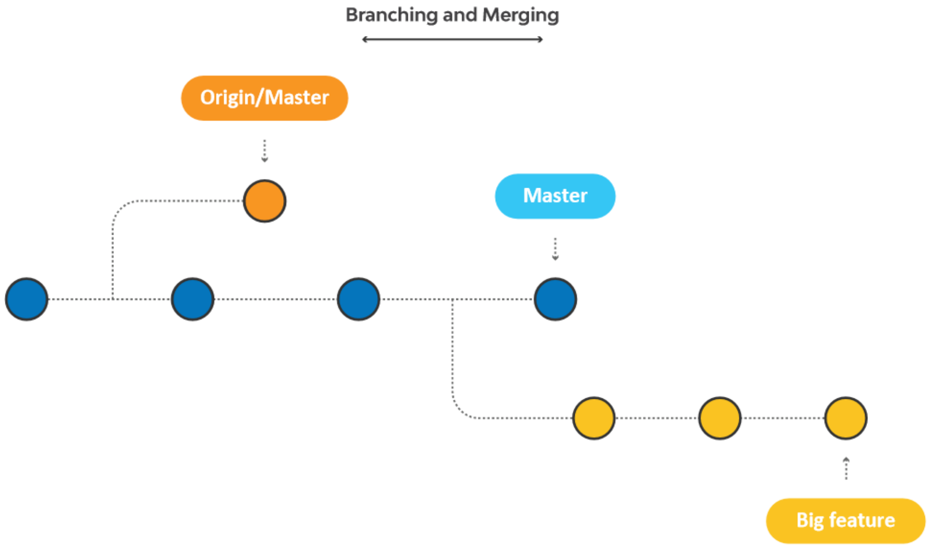
Developers make some local changes and commit these changes locally using 'git commit' command.



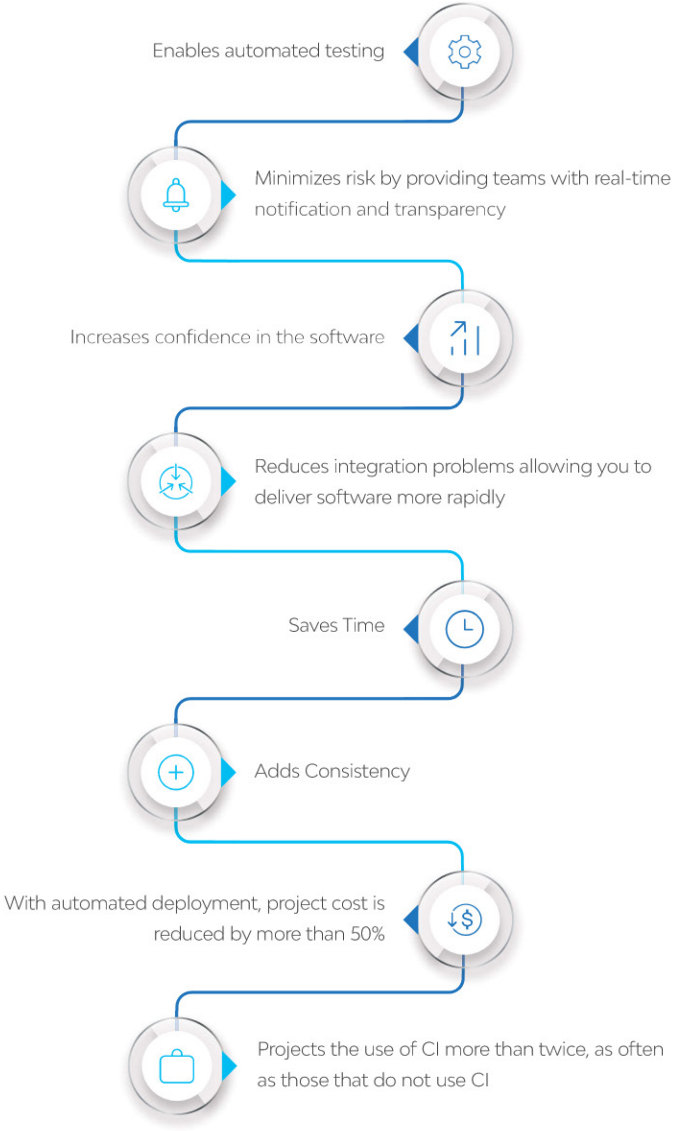
These local changes can be shared with the team and sent to the central repository or master branch, using 'git push' command as depicted in the below fig. Another team member can now retrieve the changes using 'git pull' command.



Branches are the widely used feature in Git. Developers work on their own feature branch so that they don't disturb anyone else who is also implementing some features. As shown in the given diagram Master branch is in blue, a little feature branch in purple and big feature branch in green. At some point, these branches are merged to master branch using 'git merge' command.



Key Benefits of CI



Enables automated testing

Minimizes risk by providing teams with real-time notification and transparency

Increases confidence in the software

Reduces integration problems allowing you to deliver software more rapidly

Saves Time

Adds Consistency

With automated deployment, project cost is reduced by more than 50%

Projects the use of CI more than twice, as often as those that do not use CI

Success Story

In our project, we have Bi-monthly releases which consists a lot of manual efforts for unit testing, deployment to different environments, system testing, staging deployment, etc. With implementation of CI, manual efforts are cut down by 75%, by automating complete process right from unit testing till deployment to staging.

Bug fix cost

Before CI, the team used to fix the bugs once QA testing is in progress. With implementation of CI, bugs got detected as and when the code got checked in, in the repository and helped in eliminating 75% of the development efforts.

Deployment cost

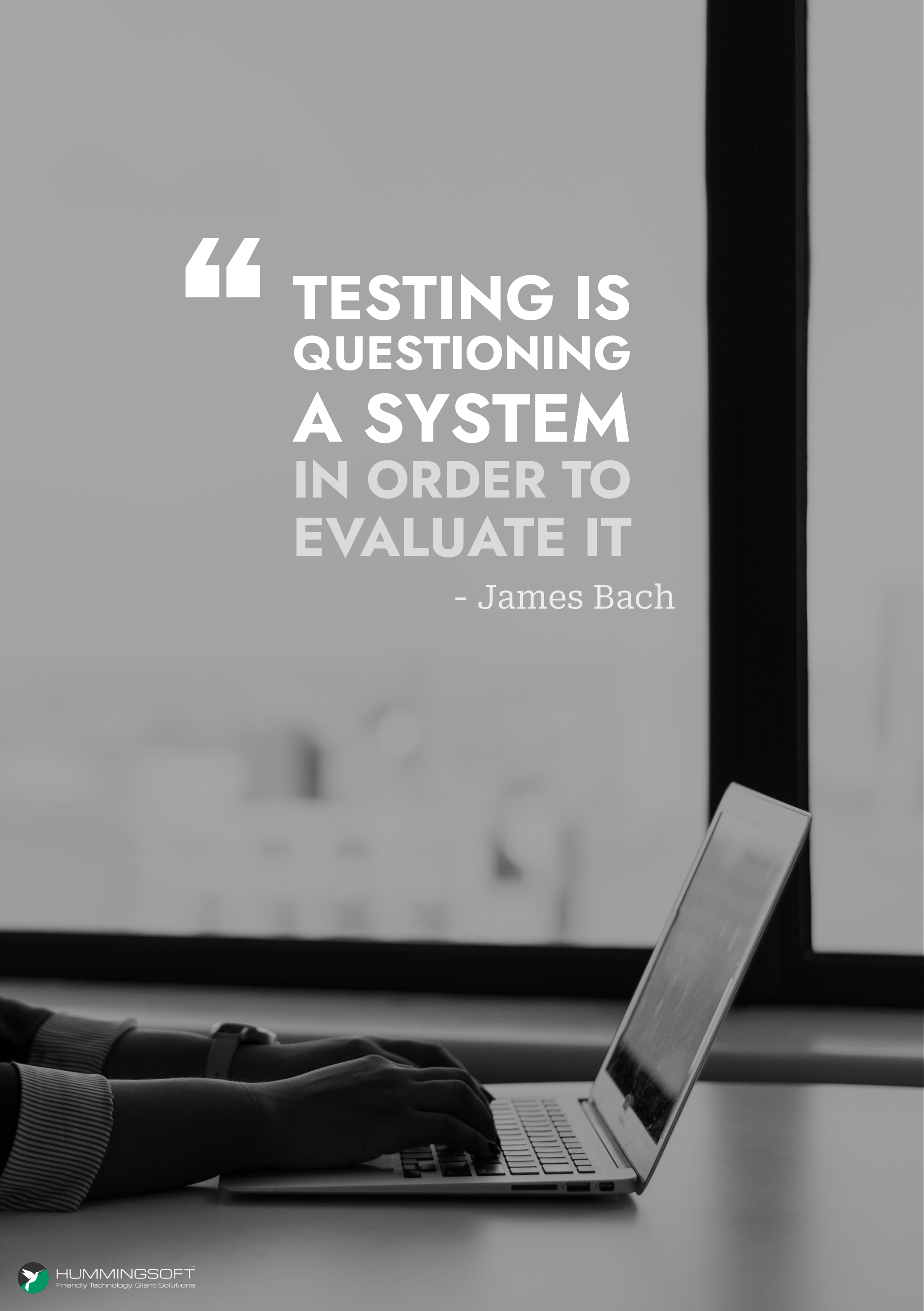
Deployment team used to take 4 hrs. for deployment per release, which is now reduced to 10 mins.

Cost of delay

Before CI, features are held on average 6 weeks after development completion before they are released. Now with CI in place features are released as and when they are ready..

Future Vision

In the future, CI will have an even greater influence than it has today. Automated environment and infrastructure creation and provisioning for cloud-based systems will mature fast. Tools like Puppet, Chef and Capistrano making automated provisioning a reality. Containers, Automated deployment, etc. are another area seeing growing adoption. Expected to have tool that grabs stack traces from production logs, and do reverse engineering of the code to auto generate the test cases in our suite. Move from Continuous Integration to Continuous Delivery, and deployment is the future.



> “TESTING IS QUESTIONING A SYSTEM IN ORDER TO EVALUATE IT
>
> \- James Bach

HUMMINGSOFT™
Friendly Technology, Giant Solutions

HUMMINGSOFT™

Friendly Technology, Giant Solutions

50-7-13A, Level 7, Wisma UOA Damansara,
50, Jalan Dungun, Damansara Heights,
50490 Kuala Lumpur, Malaysia

Tel: +60 3 2011 6558 | Fax: +60 3 2701 8873
www.hummingsoft.com.my