



magic®



CI

Continuous Integration
Continuous Improvement
of how code is written,
tested and integrated

Yuval Lavi

CTO, Magic Software Enterprises

Content

Changing Software Development: Defining CI	»	2
Key Challenges Solved by CI	»	3
The Solution Provided by CI	»	4
The CI Pipeline is a Simple, 3-Step Process	»	5
CI Advantages and Benefits	»	7
Learn More about Magic and CI	»	8

Changing Software Development: Defining CI

Since the outbreak of the COVID-19 pandemic, our lives have changed. We have all been forced to accept change, and accept it quickly. Whereas once change was considered as an integral part of improvement and growth, reality has made it an imperative fact of life. And therein lies the paradox, particularly since the propensity to change differs between every one of us.

In the world of software development, change is an ongoing and constantly evolving process that includes numerous challenges. While some developers may be skeptical at the mention of DevOps or CI, others will eagerly seek to understand how CI is going to help them improve and become more efficient.

So, what is CI? [ThoughtWorks](#) (producer of the most important articles and books on the subject) defines it as:



“... a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.”

ThoughtWorks



While CI stands for Continuous Integration in software engineering, it also represents Continuous Improvement in terms of how code is written, tested and integrated! By regularly merging working copies of code produced by developers to a shared, mainline repository several times a day, CI enables the fast detection and easy location of errors, as early as possible in the process. Consequently, CI offers the opportunity for faster and more efficient software development.

Key Challenges Solved by CI

Google “challenges faced by software engineers”. There are literally billions of results! Here we will just relate to the key challenges that are making the adoption of CI a mandatory step today:

1

Decentralization of Software Development

The production of safe, bug-free software is constantly being challenged by the ever-growing use of branching and merging, as well as the implementation of open source code produced by hundreds and even thousands of developers.

2

Business Demands

In today's digital world, businesses demand fast changes, as they constantly seek to reduce time-to-market and maintain a competitive edge. The delivery of very fast software solutions forces software engineers to simultaneously work on multiple issues.

3

Bug Detection

Detection and fixing of bugs before they escalate from minor to critical is essential. A bug detected during integration is usually a minor bug. But if the same bug is detected in production, it becomes a critical bug, requiring the investment of much more resources, under great stress, in order to fix.

4

Integration Requires More Time than the Actual Changes

The longer development continues on a branch without merging back to the mainline, the greater the risk of multiple integration conflicts and failures. Furthermore, the more changes existing in the repository, the more work developers have to invest before submitting their own changes. Eventually, the repository may become so different from the original baselines used by the developers that a chaotic situation - sometimes referred to as "merge hell" or "integration hell" - is reached, where the time required for integration exceeds the time it took to make the actual, original changes.

The Solution Provided by CI

CI is an integral part of DevOps trends to automate applications and data, optimize business value and streamline software development management. Verifying each integration of code entered by developers into a shared repository by creating an automated build, CI tests and generates a document trail, delivering important solutions to software engineers involved in the code development process:

- » Enables fast and early detection of bugs introduced into the code base
- » Allows effective synchronization and consolidation
- » Ensures that software is continuously ready for deployment at any given time
- » Improves the process for modifying the code of a package or app
- » Automates the integration of code changes from multiple contributors
- » Allows multiple development teams to work in tandem and easily incorporate new and improved features, enabling organizations to quickly meet the evolving needs of their market
- » Supplements version control systems with additional checks such as automated code quality tests, syntax style review tools, and more.



"Continuous Integration doesn't get rid of bugs, but it does make them dramatically easier to find and remove."

Martin Fowler, Chief Scientist, ThoughtWorks



The CI Pipeline is a Simple, 3-Step Process

PUSH

Pushing code to a shared repository (develop branch) for integration testing

TEST

Testing with an automated process that builds the app in a dedicated server to confirm and validate the integration of new code in current code

FIX

Fixing any bugs, malfunctions or anomalies detected, before continuing to add more features and Continuous Deployment (CD)

1

2

3

Use of CI in software development is a simple, 3-step process – namely push, test and fix. CI is ideal for the testing and deployment of functional software on a repeatable basis, enabling teams to reduce software development risks and costs while improving the quality of the software and time-to-market. And by implementing and utilizing CI, software developers ensure that they remain relevant by acquiring the skills essential for software development today and in the future. Consequently, CI is an integral part of today's software development landscape.



“The future is CI CD (continuous integration, continuous delivery) – smaller, more frequent releases that contribute less value individually but together, over time, create more value for all of their stakeholders from citizens and customers to firms and their suppliers.”

[Gartner, The Key to a Successful Digital Transformation](#)

Gartner



Although this process appears simple, it requires a change in organizational work culture, support from management, and the participation of all team members.

Pushing relates to the frequent pushing of the code written for a new feature to the shared repository. This demands the biggest change in work culture, because it requires software engineers to modify the way they work – to frequently push the code into the develop branch for testing. This continuous pushing triggers continuous integration testing.

Testing relies on automated, reliable tests to validate new code as it is pushed each time to the repository. It can include any number of tests that are considered critical, but the CI loop should be kept as short as possible so that feedback can be provided as quickly as possible. Naturally, this automatic detection of errors includes the receipt of relevant messages/alerts, thereby ensuring speed, increasing productivity and eliminating human error.

When a build is broken, **fixing** it is the top priority for the development team. This is because no-one can continue pushing code changes until the fix is complete. And of course, if the build cannot be fixed quickly, the team can decide to remove the code. The principle here is that the build will always produce working code that can be released.

An Analogy to CI: Development of New Part for a Vehicle

Consider the development of a new wheel for an existing vehicle. Similar to a new software feature, the new wheel consists of a number of parts, such as the hub, spokes, rim and tire. So, when development of the new wheel is complete, it has to be PUSHED to the manufacturing line to be tested. This involves a range of integration tests, from whether the wheel can actually be connected to the vehicle and turn to whether other interrelated vehicle systems – such as brakes, air pressure sensors, wipers and warning lights - are still operational with the new wheel.

In this scenario, PUSH is pushing the vehicle part to the manufacturing line. TEST is testing the wheel with static integration tests. And FIX is the fixing of problems detected, such as if the brakes do not work, there is an integration problem. Once a fix has been performed, the cycle starts again from PUSH.

The significance is clear. Optimal testing of the wheel and elimination of bugs and malfunctions is performed during and immediately after the vehicle assembly process, thereby preventing much more serious problems down the line. Similarly, a new software feature reaching production with a bug makes it much more difficult to locate the source of the bug.



CI Advantages and Benefits

Martin Fowler once stated that one of his favorite soundbites was *"if it hurts, do it more often"*. He explains the relevance particularly in relation to integration testing. By testing more frequently, it is possible to reduce the pain in development. By using CI to integrate (push) new builds every day, the pain of integration is significantly reduced.

As an important and integral part of DevOps today, CI offers freedom for developers and peace of mind for management.

Freedom for Developers

Early detection and tracking of integration bugs, eliminating long and tense integrations

Compatibility between versions

Frequent committing of code for integration tests prevents integration problems; only a few changes lost when reverting back to a bug-free state

Focus on developing functional, quality code, and supporting the development of team momentum

Dedicated integration build machine using continuous feedback mechanisms

Less time testing and debugging

Immediate feedback on system-wide impact of local changes

Peace of Mind for Management

Saves time and money in project lifecycle

Avoids last-minute chaos at release dates

Pushes developers to create modular, less complex code

Awareness of project status through automatic generation of metrics, such as code coverage, complexity and features completed

Better visibility and communication

More time available for adding features

Constant availability of a "current" build for testing, demo, or release purposes



"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency."

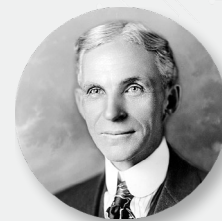
Bill Gates



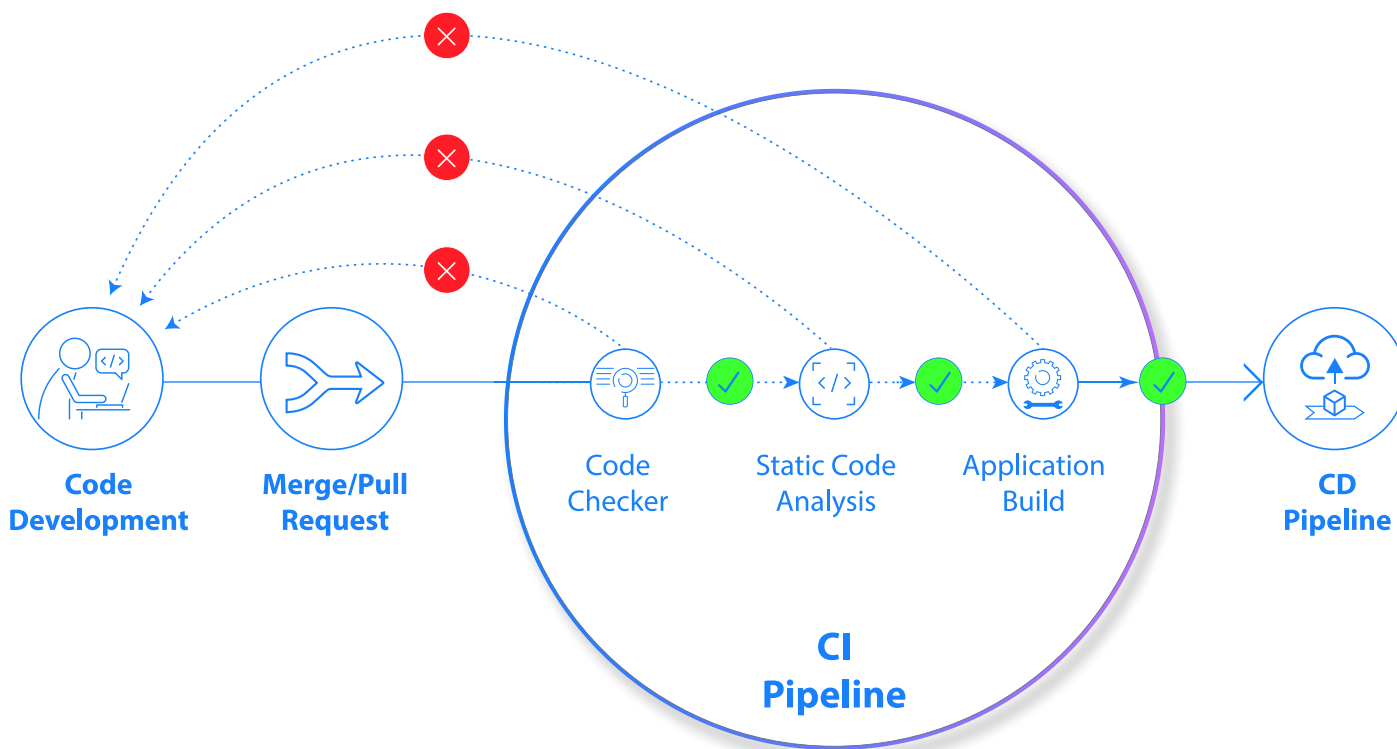
Learn More about Magic and CI

*“If you always do what you’ve always done,
you’ll always get what you’ve always got.”*

Henry Ford



CI offers the opportunity to transform development efforts and quickly, safely and reliably deliver software to a production environment. The change involved delivers both the freedom demanded by developers and the peace of mind needed by management.



With this in mind, Magic is integrating CI/CD methodology in its current and future roadmap. This will enable Magic xpa platform developers to take advantage of CI and improve the way they write, test and integrate code. Based on a Git repository that enables the build and integration testing of all new code, typical Magic CI pipeline workflows include:

- » Code checks (code compilation and application checker in Magic xpa)
- » Static code analysis
- » Application and component builds

Want to learn from our experts how CI is being used in Magic development environments?

Contact us for more details or to schedule a demo



About Magic

With over 35 years of experience, Magic Software has established its position in the market as a global software provider, present in 24 regional offices, with thousands of installations worldwide, and strategic alliances with global IT leaders, such as IBM, Microsoft, Oracle, Salesforce, and SAP.

Magic Software is part of the Magic Group (NASDAQ and TASE: MGIC), a global provider of proprietary application development and business process integration software solutions and related professional services, and a vendor of a variety of IT professional services.

The Magic team collaborates closely with our customers and over a thousand business partners in over 50 countries, as we believe that our community is our biggest asset.

We welcome all of our customers, developers, partners, investors and prospects to learn more about our solutions and invite you to take part in our continuous journey to help companies digitally transform, making the most out of their data and technology.



About the author

Yuval Lavi joined Magic in June 2013 with the mission of providing high-level professional services. After three years as the Global VP of Professional Services, Yuval moved to the position of CTO. In his new position, Yuval is in charge of defining the company's technological vision, opening new technology horizons for the company and building strategic technical alliances.

