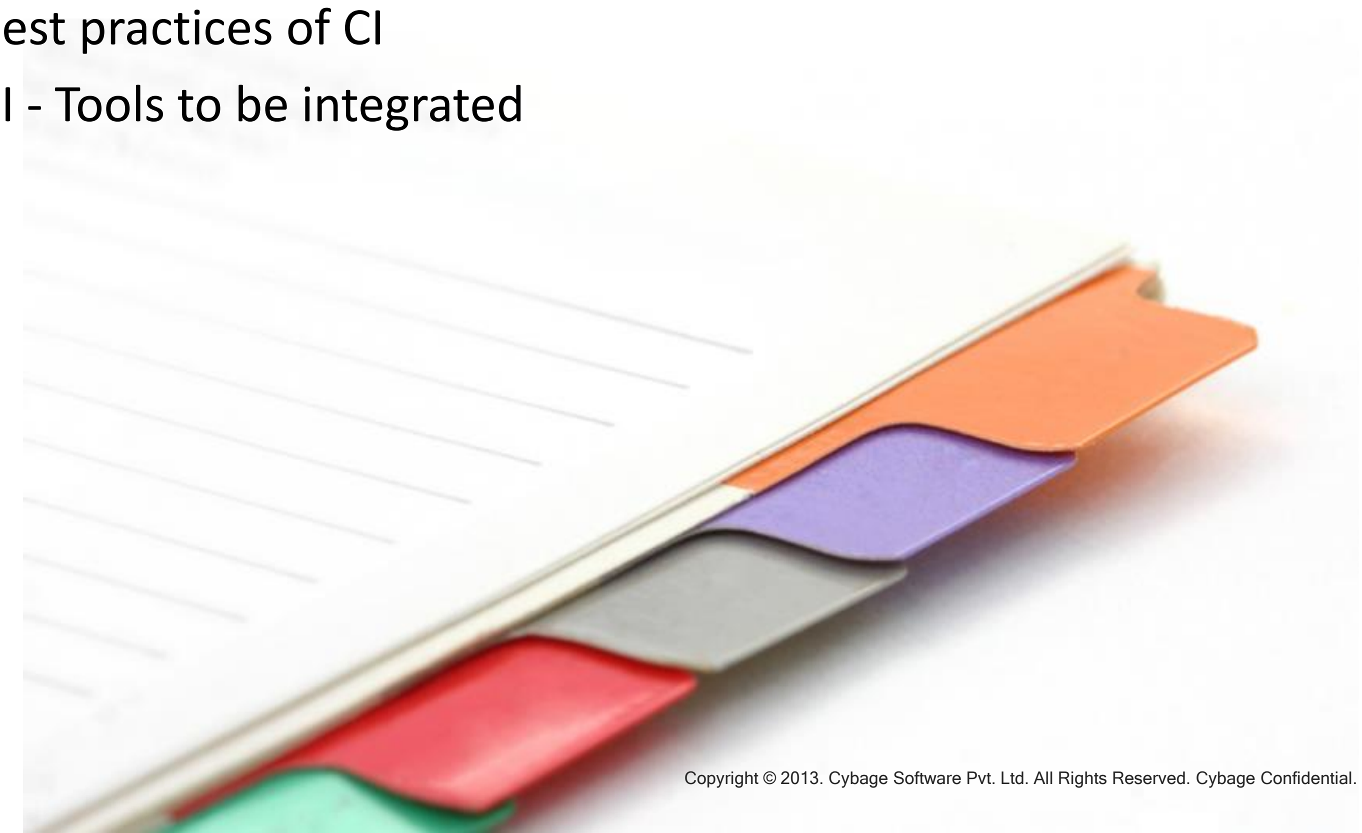
Continuous Integration

Authored and Presented by : Cybage Software Pvt. Ltd.

This presentation is the intellectual property of Cybage Software Pvt. Ltd. and is meant for the usage of the intended Cybage employee/s for training purpose only.
This should not be used for any other purpose or reproduced in any other form without written permission and consent of the concerned authorities.

Agenda

- What is Continuous Integration
- Benefits of CI
- How does CI work?
- Best practices of CI
- CI - Tools to be integrated



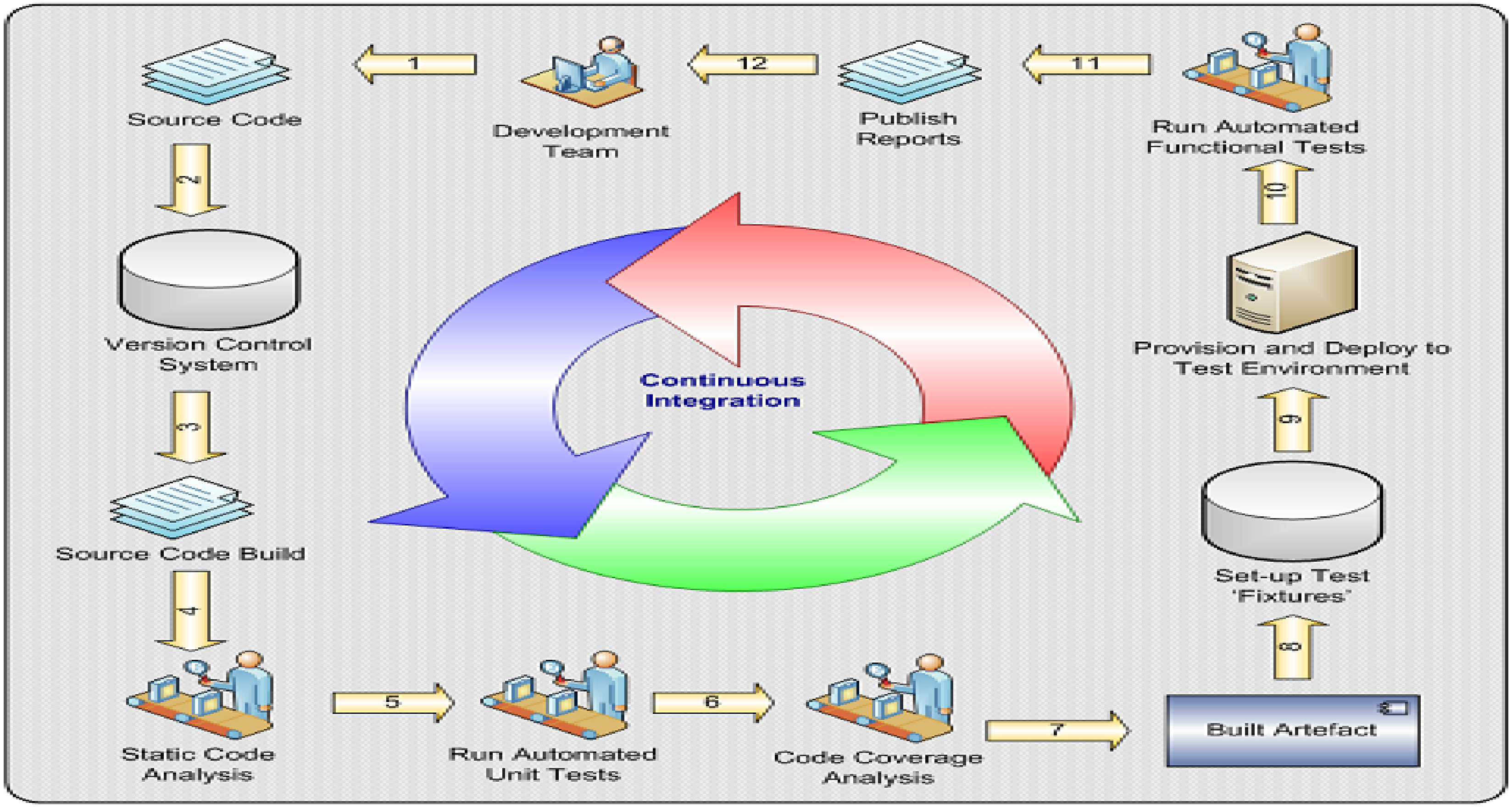
What is Continuous Integration

- Continuous Integration (CI) is the process of building software with every change committed to a project's version control repository.
- CI is a practice of constantly merging source code and building/testing as often as possible.
- Continuous integration involves integrating early and often, so as to avoid the pitfalls of "**integration problems**".
- The practice aims to reduce rework and thus reduce “cost and time.”
- Development process automation technique.

Benefits of CI

- **Increase visibility** which enables greater communication
- **Spend less time** debugging and more time adding features
- **Reduce integration** problems allowing you to deliver software more rapidly
- Reverting to a **bug-free state** in case of build failures
- **Avoid** last minute chaos at release dates
- **Immediate unit** and **intégration testing** of all changes
- **Limit the risk** of regression
- **Immediate feedback** to developers on the quality, functionality, or system-wide impact of code they are writing
- **Metrics** generated from automated testing and CI focus developers on developing functional, quality code

How does CI work?



Important CI practices

- Maintain a code repository
- Automate the build
- Make your build self-testing
- Every commit should build on an integration machine
- Test in a clone of the production environment
- Make it easy for anyone to get the latest executable

How to achieve CI?

- Developers check out code into their private workspaces.
- When done, they commit changes to the repository.
- CI server monitors Repository & checks out changes when they occur.
- The CI server builds the system and runs unit and integration tests.
- The CI server releases deployable artifacts for testing.
- CI server assigns a build label to the version of the code it just built.
- CI server informs the team of the successful build.
- If the build or tests fail, the CI server alerts the team.
- The team fix the issue at the earliest opportunity.
- Continue to continually integrate and test throughout the project.

Team Responsibilities in CI

- Check in frequently
- Don’t check in broken code
- Don’t check in untested code
- Don’t check in when the build is broken
- After checking in ,make sure the system builds

Thank You!