

Continuous Integration



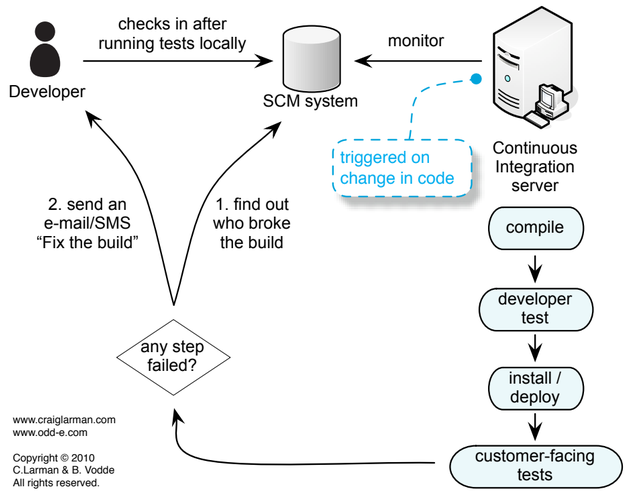
Dario Campagna

Head of Research

Continuous Integration

A software development practice to detect integration errors as quickly as possible.

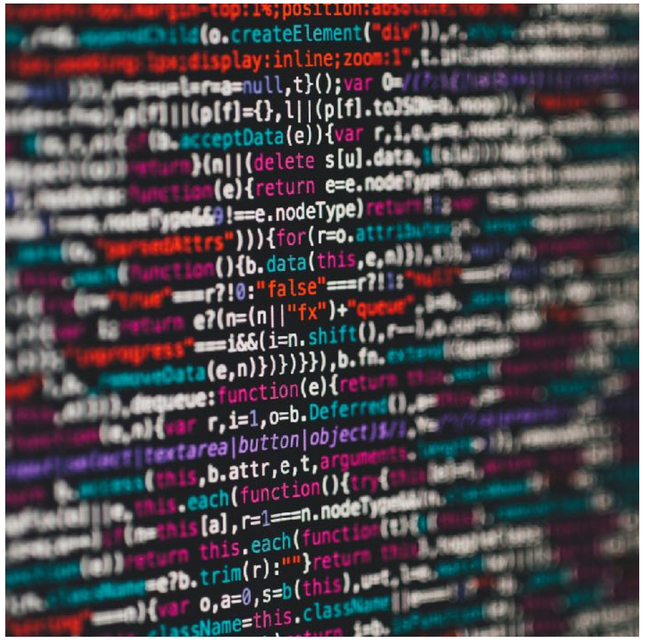
- Members of the team integrate their work frequently
- Multiple integrations per day
- Automated verification of each integration



Software Systems: common issues

Software systems are complex.

- A change to a single file can break the system.
- Combining the work of multiple developers is hard.



Software Systems: usual solution

Developers work on their own branches.

- To keep trunk/main stable.
- To prevent treading on each other toes.

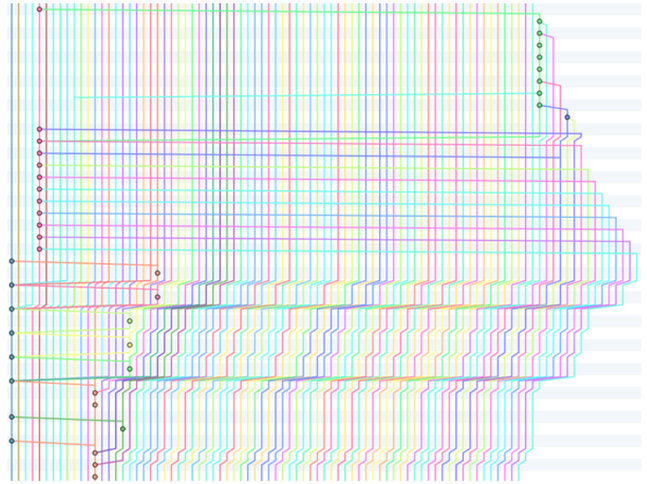


Image from https://www.freecodecamp.org/news/why-you-should-not-use-feature-branches-a86950126124/

Long-lived Branches

Issues

Painful to integrate into mainline

Need of code freezes, integration and stabilization phases

Expensive and unpredictable process

Issues become more sever as

Team sizes grow

Branches become more long-lived



Continuous Integration to the rescue

XP principle: if something is painful, we should do it more often, and bring the pain forward.

- Developers integrate all their work into trunk regularly
- Automated tests are run before and after the merge
- If automated tests fail, stop and fix immediately

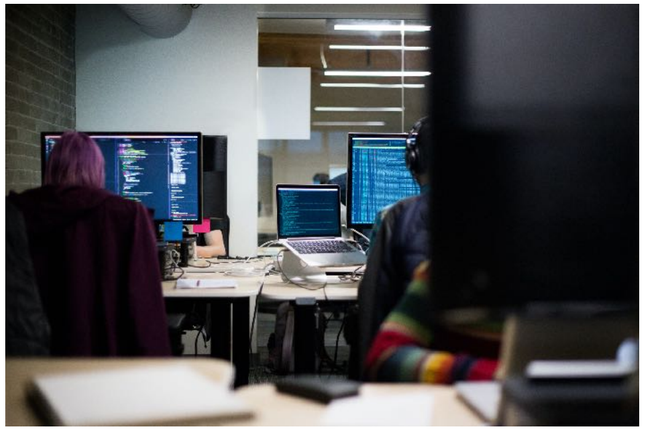


Mustafasari, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2816241

Developer practice

Continuous integration is a practice–it is about what people do, not about what tools they use.

- Adopting CI requires a change in human behavior.
- CI requires a change to the daily habits of developers.



Keep a working system

Continuous integration helps in keeping a working system.

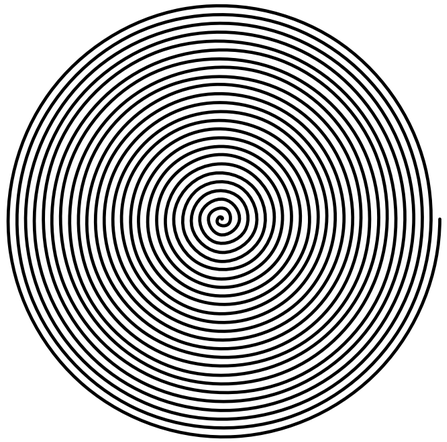
- CI means always having a stable system.
- When a test fails the developer fixes it immediately.
- CI increase visibility by removing un-integrated code.



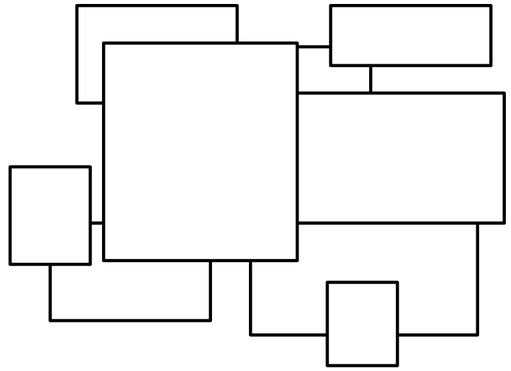
Growing the system

Continuous Integration means to grow the system.

- Building: separate components are assembled together when they are finished.
- Growing: nurturing the system and evolving it into a larger system.



growing



building

www.craiglarman.com
www.odd-e.com

Copyright © 2010
C.Larman & B. Vodde
All rights reserved.

Small changes

Continuous integration requires small changes.

- Integrate in the system easily.
- Integrate in the system fast.
- Developers get regular feedback on the impact of their work.



Integrate at least daily

How frequent is ‘continuous’? As frequently as possible! This is limited by…

- Ability to split large changes.
- Speed of integration.
- Speed of feedback cycle.



On the mainline

We want to integrate on the mainline.

- You may still need releases branches.
- Very short-lived branches are fine.
- Feature toggles, branch by abstraction.
- More about branch management here.

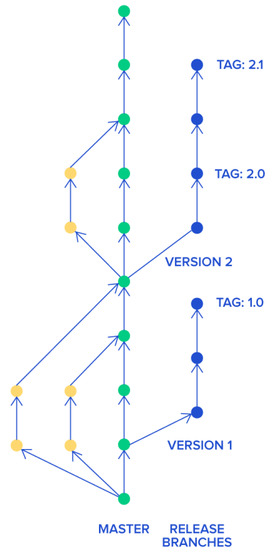
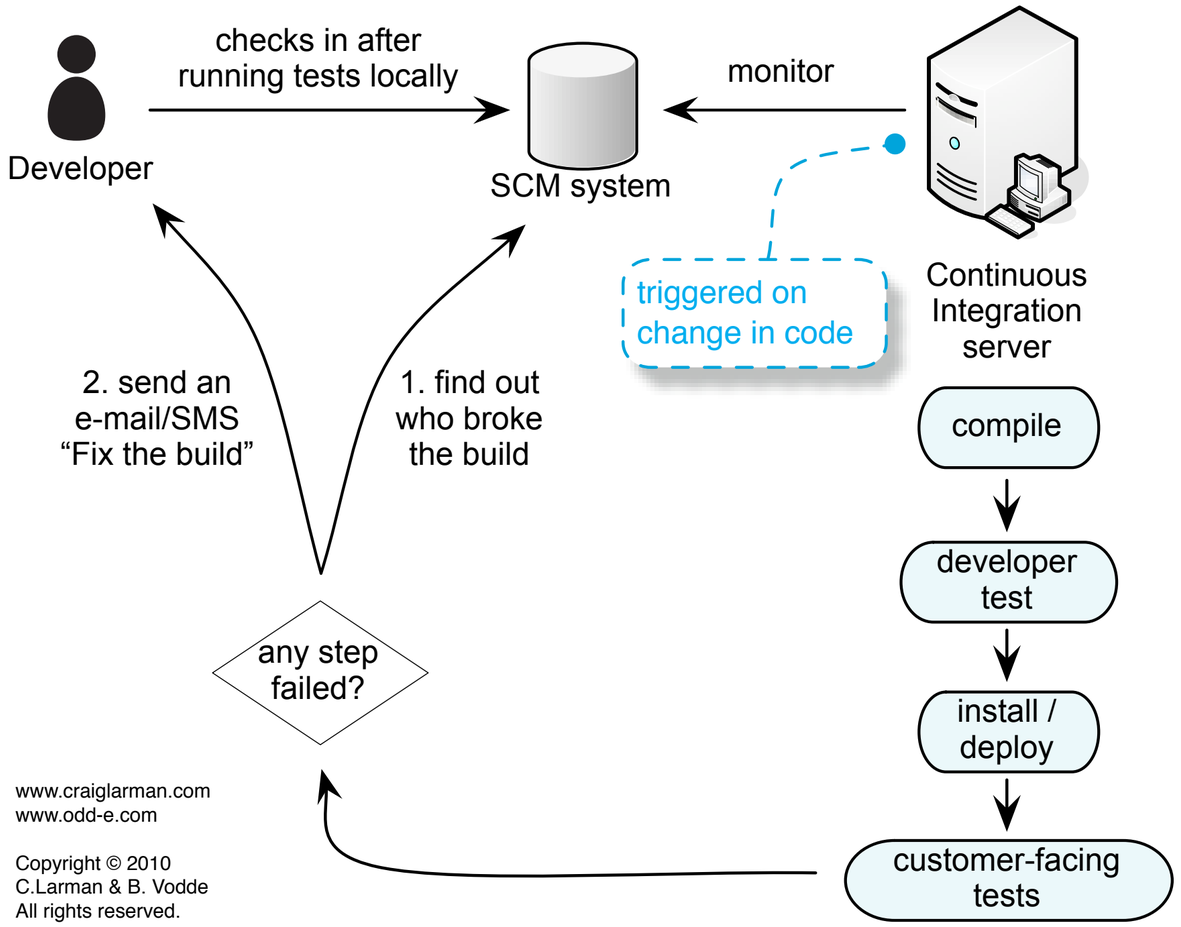


Image from https://www.toptal.com/software/trunk-based-development-git-flow

With the help of a CI system



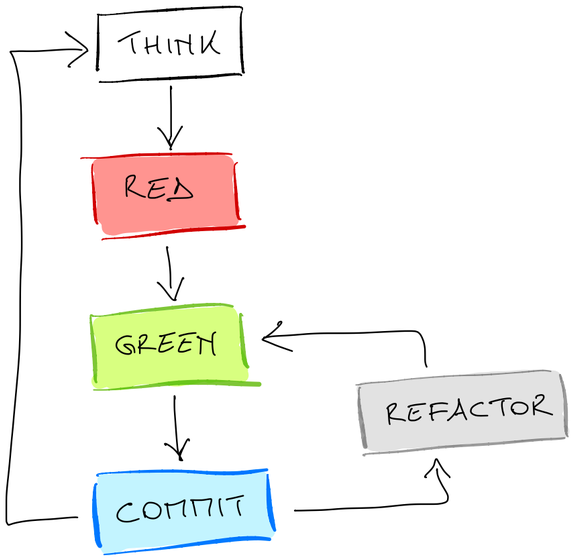
www.craiglarman.com
www.odd-e.com

Copyright © 2010
C.Larman & B. Vodde
All rights reserved.

With lots of automated tests

Continuous Integration requires you to have lots of automated tests.

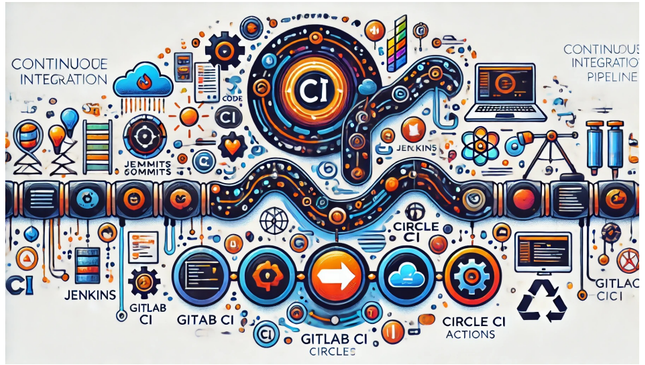
- To have a CI system compile everything is not very useful.
- More automated tests means better safety net.
- More automated tests means more confidence the system is working.



Examples of tools to implement a CI system

Different tools are available nowadays, from open-source automation servers to cloud platforms.

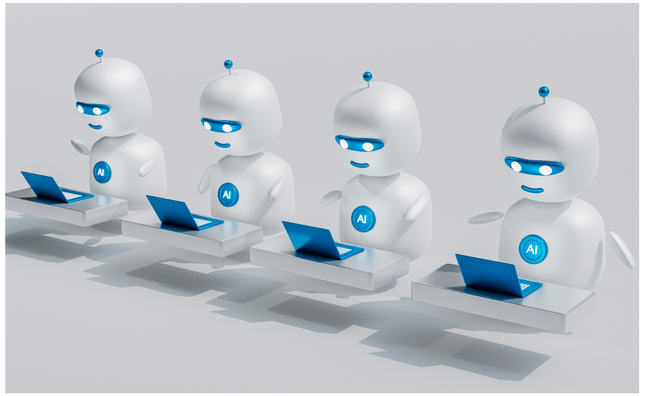
- Jenkins
- Circle CI
- Travis CI
- GitHub Actions
- GitLab CI/CD



CI and coding agents

CI becomes even more central with coding agents.

- CI becomes the agent’s feedback loop, not just a safety net.
- Volume and parallelism stress the integration part.
- Pipeline speed becomes a bottleneck.
- The verification system needs to be protected from the agents.



References



Continuous Integration
Martin Fowler

Continuous Delivery
Jez Humble, David Farley

Accelerate
Nicole Forsgren Ph.D., Jez Humble, Gene Kim