Continuous Integration

Is not automating the build and running tests

Continuous Integration is…

…a **software development practice** where members of a team **integrate** their work **frequently**, usually each person integrates **at least daily** - leading to multiple integrations per day. Each integration is **verified** by an **automated build** (including **test**) to **detect integration errors as quickly as possible**.

Martin Fowler

Common Issues

- Combining the work of multiple developers is hard

- Software systems are complex

- A change to a single file can break the system

Usual Solution

- Developers work on their own branches

- To keep trunk/master stable

- To prevent treading on each other toes

Long-lived Branches

- Painful to integrate into mainline

- Code freezes, integration and stabilization phases

- Expensive and unpredictable process

With Continuous Integration

- 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

Continuous Integration is

- A developer practice…

- To keep a working system

- By small changes

- Growing the system

- By integrating at least daily

- On the mainline

- Supported by a CI system

- With lots of automated tests

Developer Practice

- It is about what people do, not about what tool they use

- Adopting CI requires a change in human behavior

- CI requires a change to the daily habits of developers

Keep a Working System

- CI means always having a stable system

- When a test fails the developer fixes it immediately

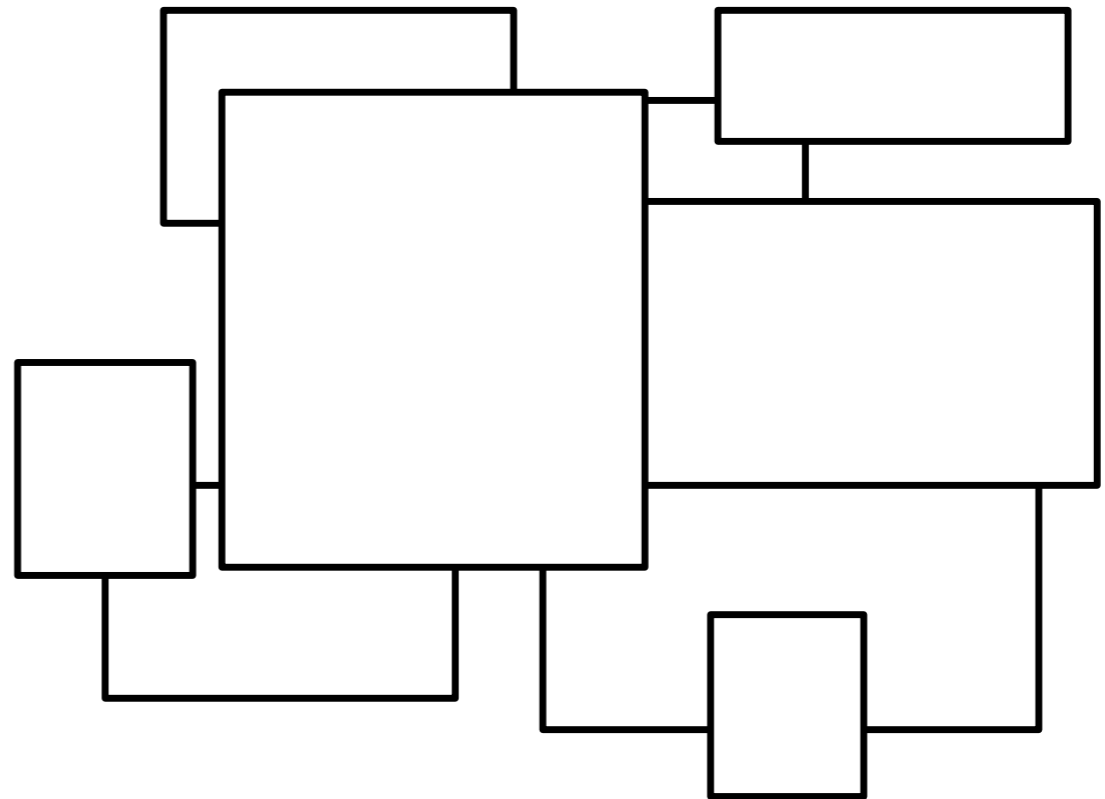- CI increase visibility by removing un-integrated WIP

Small Changes

- CI requires developers to break up large changes

- CI needs small, incremental changes

- Large changes can brake a system in large ways

- Small changes integrate in the system easily

Growing the System
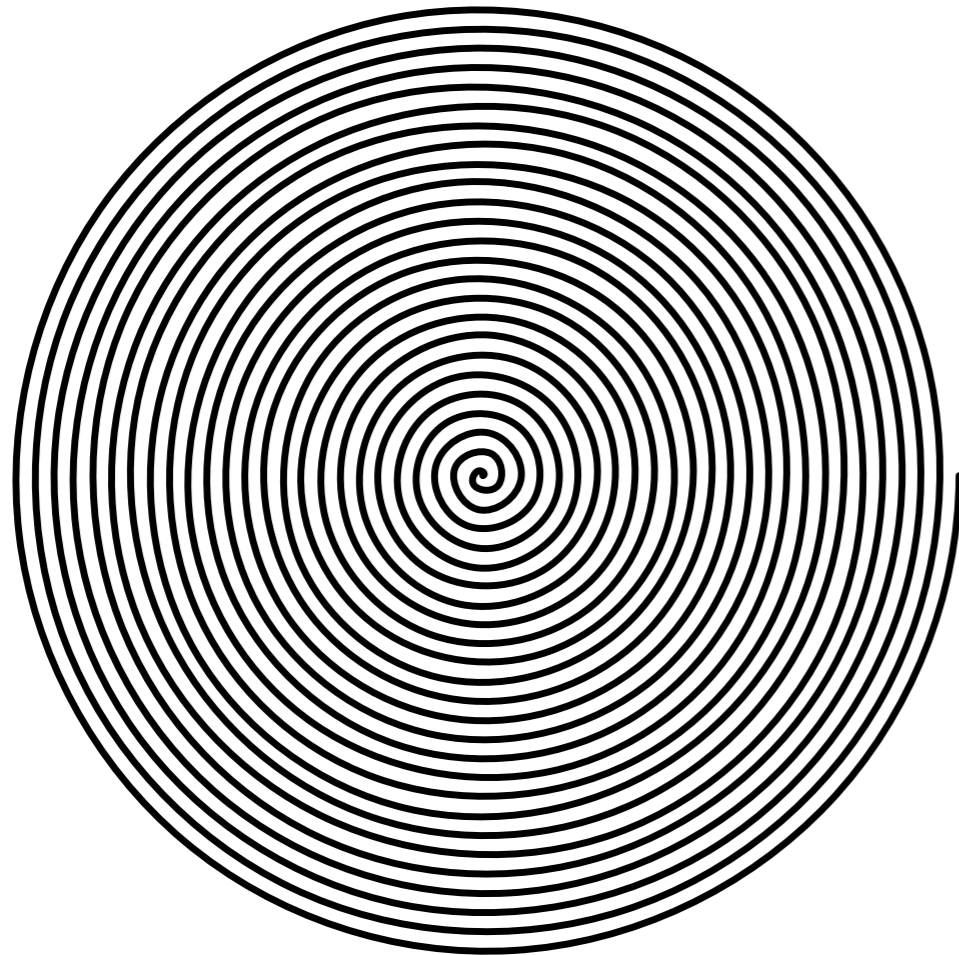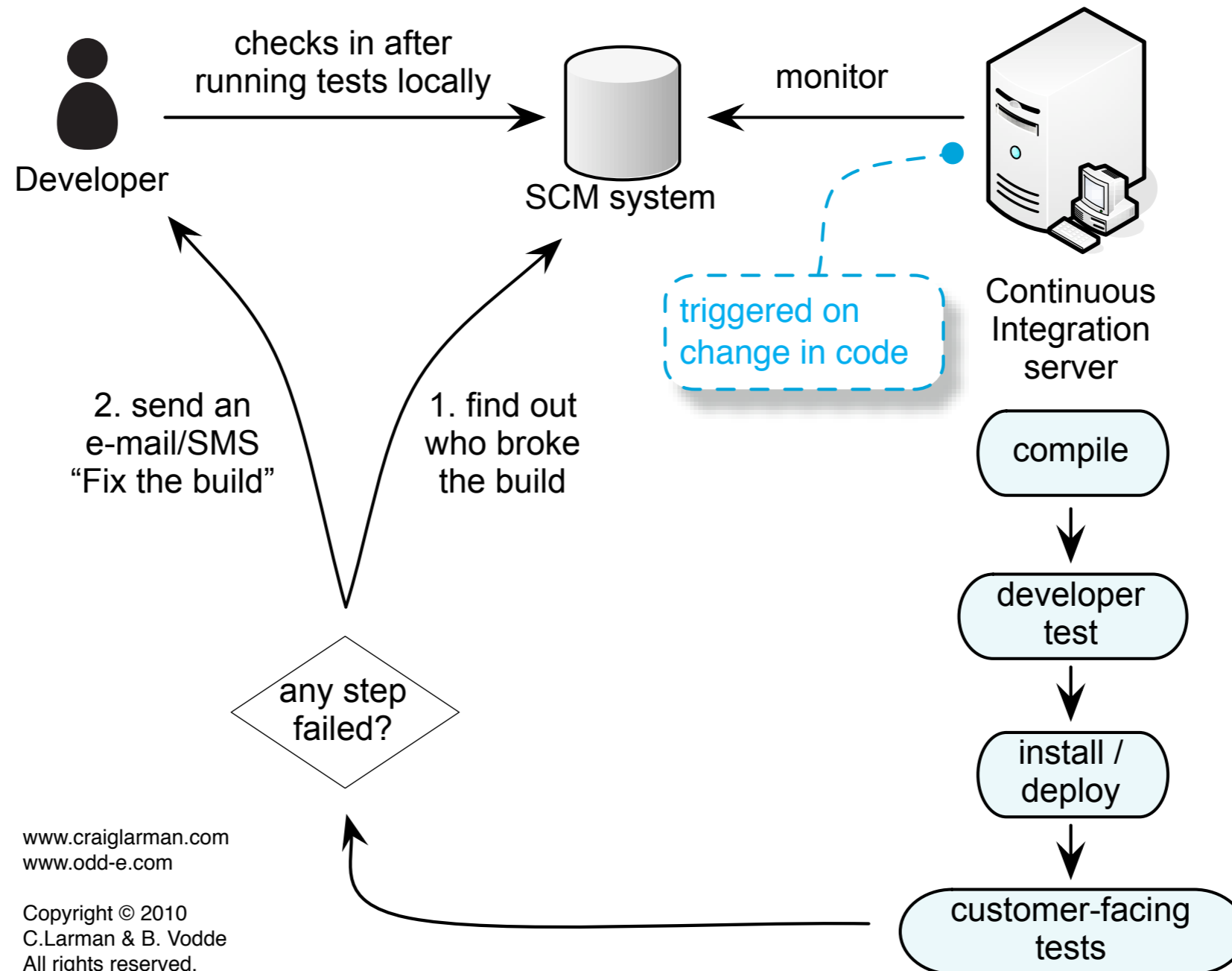
growing

building

At Least Daily

- Ability to split large changes

- Speed of integration

- Speed of feedback cycle

On the Mainline

- No long-lived branches

- Release branches may be needed

- Feature toggles, branch by abstraction

Supported by a CI System

checks in after
running tests locally

Developer

SCM system

monitor

triggered on
change in code

Continuous
Integration
server

compile

developer
test

install /
deploy

customer-facing
tests

2. send an
e-mail/SMS
"Fix the build"

1. find out
who broke
the build

any step
failed?

Lots of Automated Tests

- To have a CI system compile everything is not very useful

- The more automated tests

  ‣ The better your safety net

  ‣ The more confidence you system is working

How to start?

- Change developer behavior

- Set up a CI system