Trunk-Based Development

Concepts and Experiences from trenches

Prof. Eng. Nicolás Paez

@inicopaez

Context

http://nicopaez.com.ar/tbd-survey

Trunk-based Development?

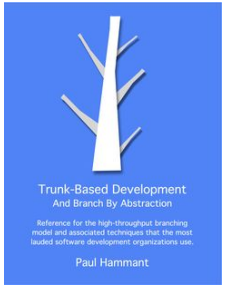
Don’t know

Don’t trust

Don’t need

What is Trunk-Based Development?

Trunk-based Development



A source-control branching model, where developers collaborate on code in a single branch called ‘trunk’, resist any pressure to create other long-lived development branches by employing documented techniques. They therefore avoid merge hell, do not break the build, and live happily ever after.

Paul Hammant

Why is it important?

Trunk-based Development

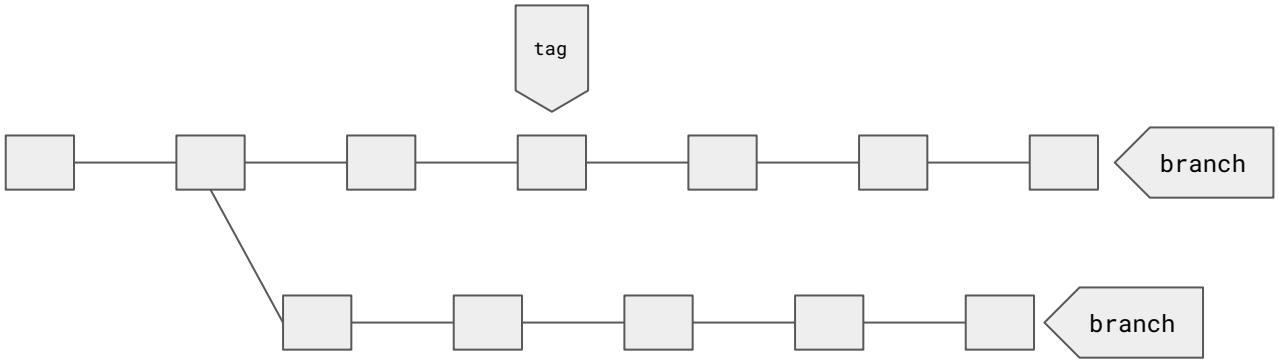
It is required for CI/CD

It makes life easier

Vocabulary

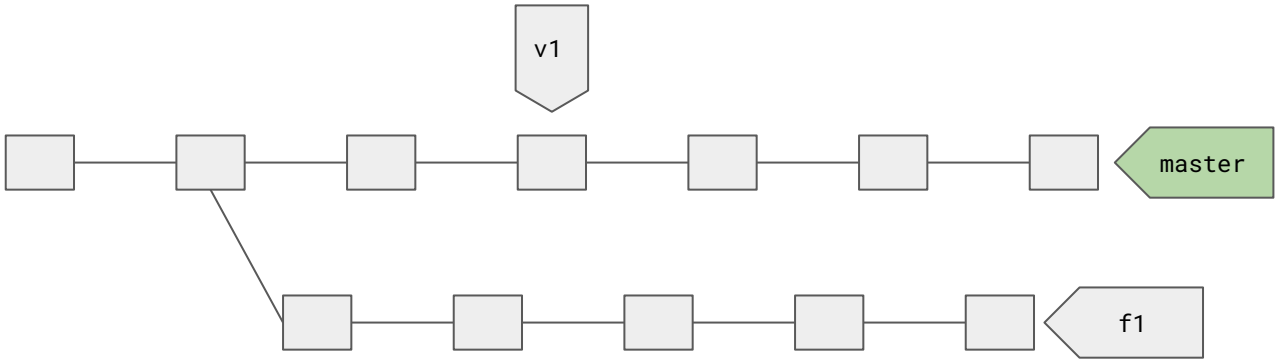
Codeline

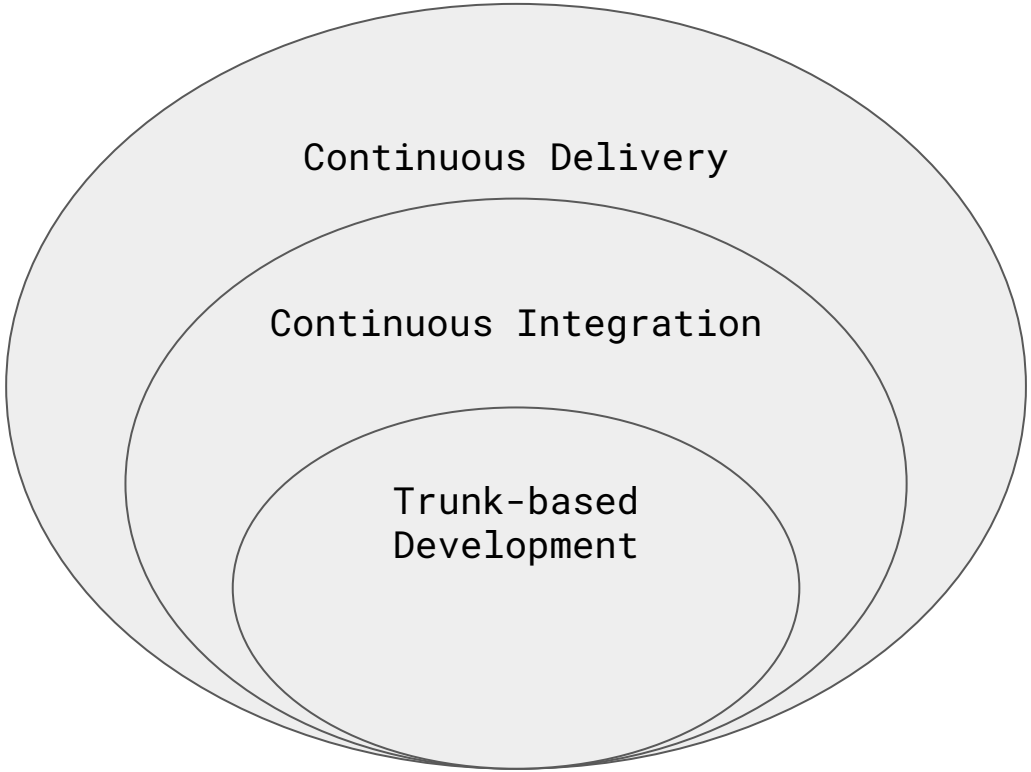
A particular sequence of versions of the code base. It can end in a tag, be a branch, or be lost in git's reflog.

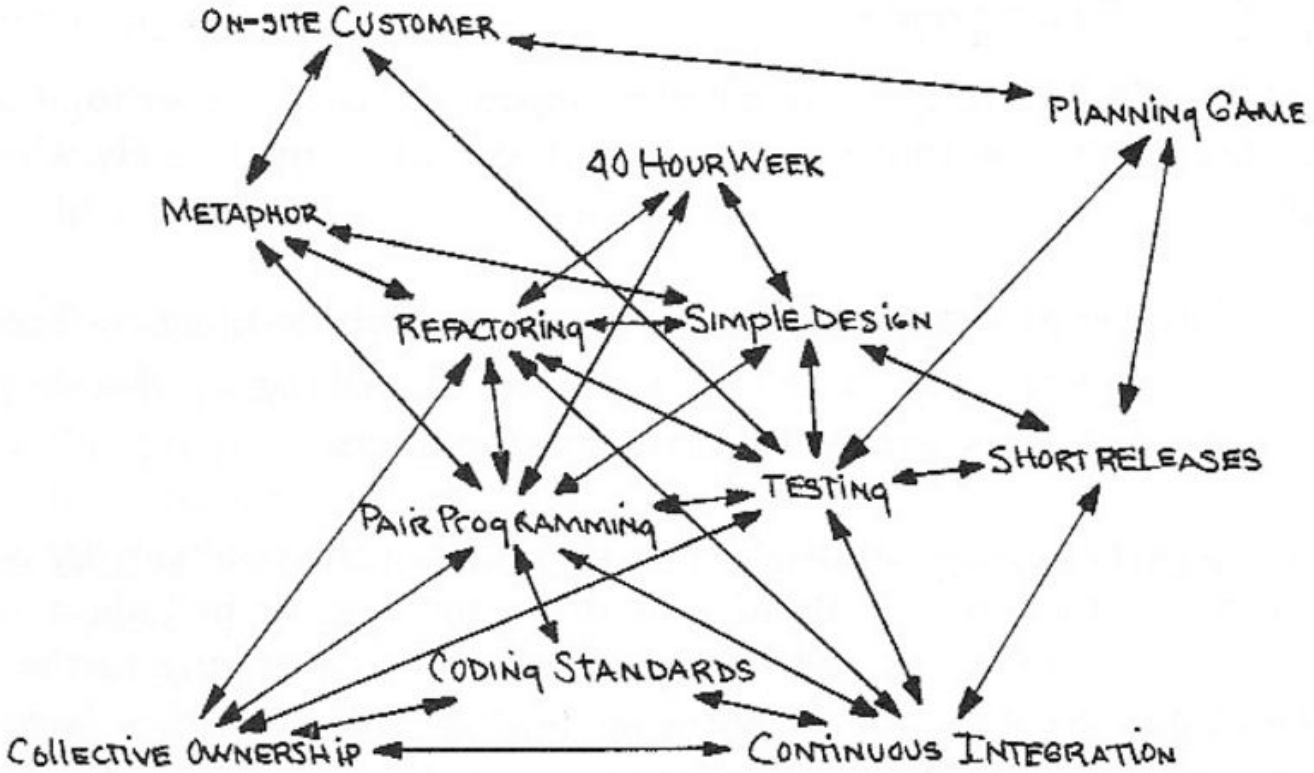


Mainline

A single, shared, branch that acts as the current state of the product




Continuous Delivery
Continuous Integration
Trunk-based
Development



Trunk-based development

Red de Prácticas de XP

Continuous Integration

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. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. This article is a quick overview of Continuous Integration summarizing the technique and its current usage.

01 May 2006

Martin Fowler

POPULAR

AGILE

CONTINUOUS DELIVERY

EXTREME PROGRAMMING

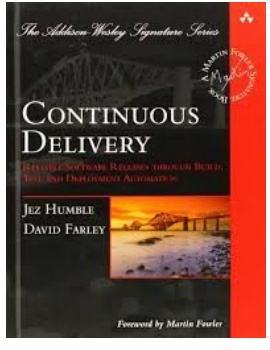
CONTENTS

Building a Feature with Continuous Integration
Practices of Continuous Integration
Maintain a Single Source Repository.
Automate the Build
Make Your Build Self-Testing
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
Test in a Clone of the Production Environment
Make it Easy for Anyone to Get the Latest Executable
Everyone can see what's happening
Automate Deployment
Benefits of Continuous Integration
Introducing Continuous Integration
Final Thoughts

https://martinfowler.com/articles/continuousIntegration.html
Martin Fowler

If different members of the team are working on separate branches or streams, then by definition they're not continuously integrating. Perhaps the most important practice that makes continuous integration possible

So if you merge your branch to (not just from) mainline once a day, you're OK. If you're not doing that, you're not doing continuous integration. Indeed, there is a school of thought that any work on a branch is, in the lean sense, waste—inventory that is not being pulled into the finished product.



Release != Deploy

Challenges
(myths / doubts)

Branch usage

Parallel development

Spikes / PoC

Environment management

Merge conflicts

Merge-Requests

Techniques / Strategies

Feature toggles

```
[FeatureGate(Features.Pagos)]
public ActionResult PagarConCuenta(string cuentaCredito)
{
    // ....
}
```

```
if (FeatureManager.IsFeatureEnabled(Features.Pagos, context))
{
    // ....
}
```

```
"FeatureManagement": {
  "MailPagos": false,
  "Pagos": {
    "EnabledFor": [
      {
        "Name": "Targeting",
        "Parameters": {
          "Audience": {
            "Users": [
              "john.doe",
              "juan.perez",
              "larry.smith"
            ]
          }
        }
      }
    ]
  }
```

XP Practices

Small Teams => 8 Devs => 8 branches

Pair-Programming => 4 branches

Mob-Programming => 1 branch

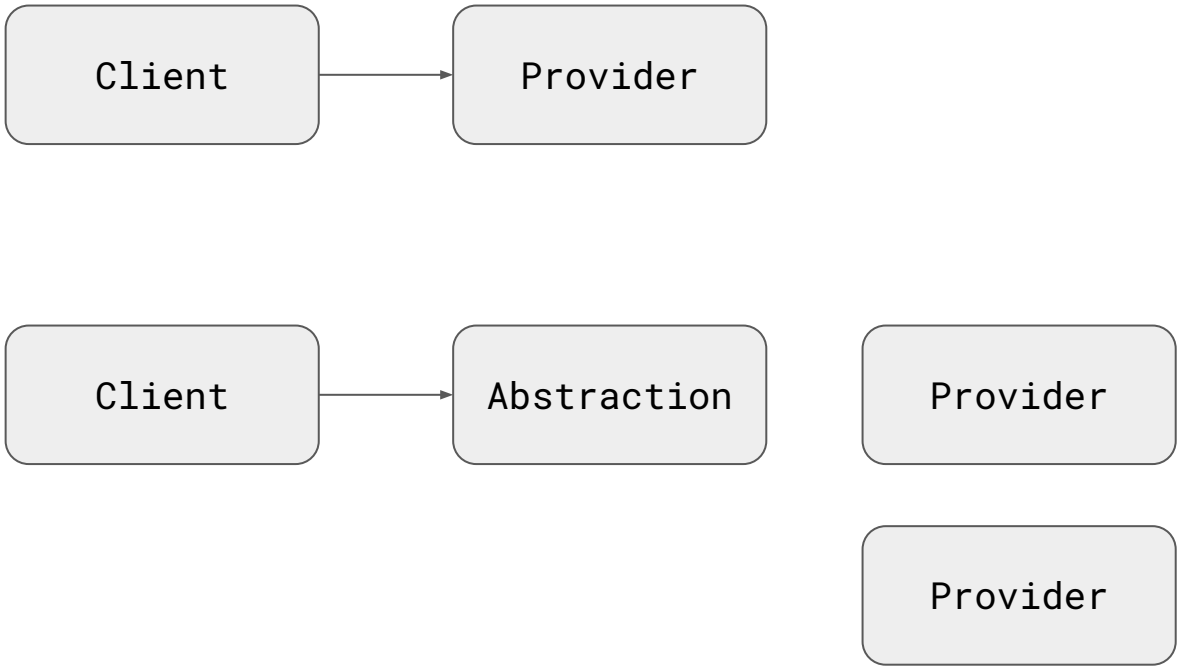
Roll-forward

Branch for release

vs.

Release from trunk

Strangulation
&
Branch by Abstraction



Branch by Abstraction

How I practice it

One branch for development & releasing

Pair/Mob Programming all time

for production code

BDD / TDD
(london style)

Semantic version + commit-id embedded in binary

```
GET /api/health => { “version”:”1.0.35-6ef4976”}
```

Resources

https://blog.nicopaez.com/2020/03/20/todos-contra-master-trunk-based-development/

https://trunkbaseddevelopment.com/book/

https://blog.nicopaez.com/2020/07/05/continuous-delivery-como-una-cinta-transportadora/

https://www.youtube.com/watch?v=ZnVMsZX3WU0

https://martinfowler.com/articles/branching-patterns.html

The End

Nicolás Paez
nicopaez@computer.org
@inicopaez
blog.nicopaez.com