BRANCHING PATTERNS

LUIS A. FERNANDEZ SUAREZ

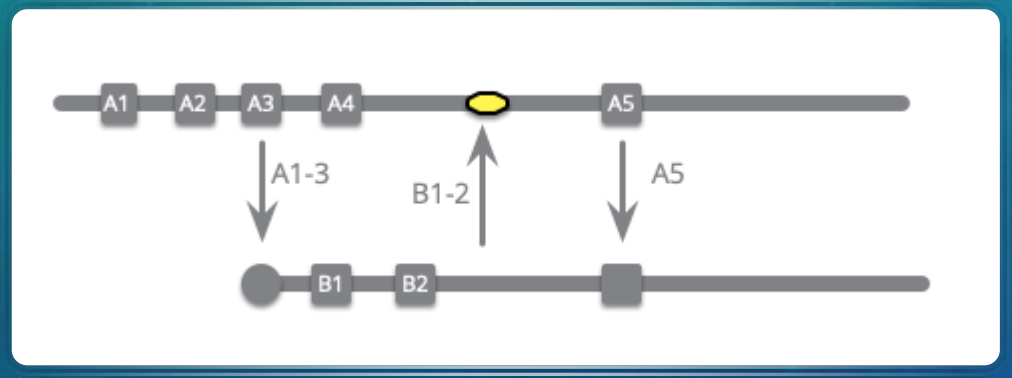
KAAN IPEK

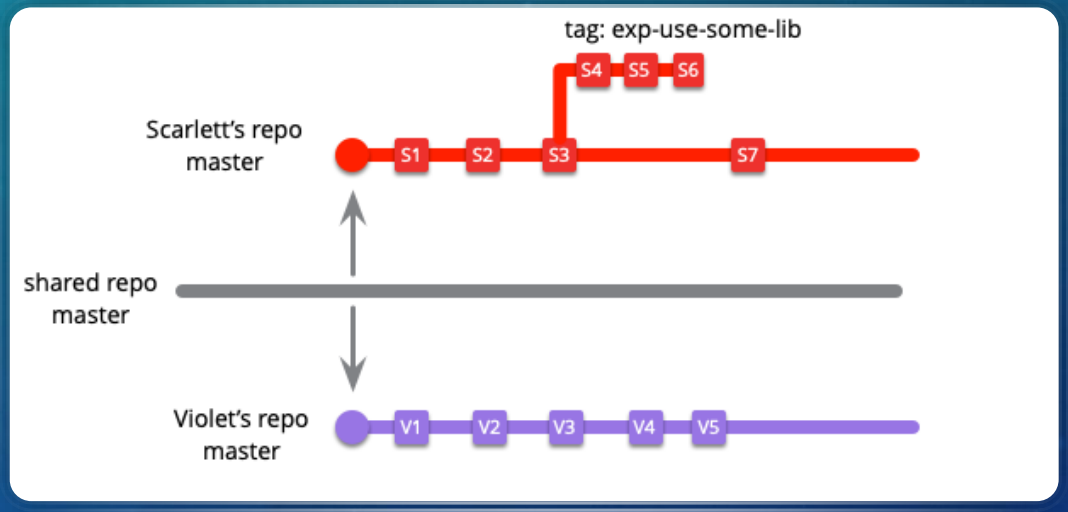
MIGUEL LIGERO ARBESÚ

CARMEN RENDUELES MARTÍNEZ

BASE PATTERNS

SOURCE BRANCHING





MAINLINE

"A SINGLE, SHARED, BRANCH THAT ACTS AS THE CURRENT STATE OF THE PRODUCT"

HEALTHY BRANCH

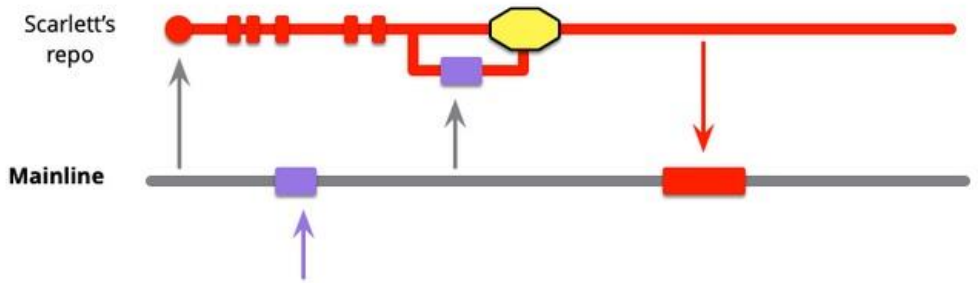
Self Testing Code

Directly implementation

More production

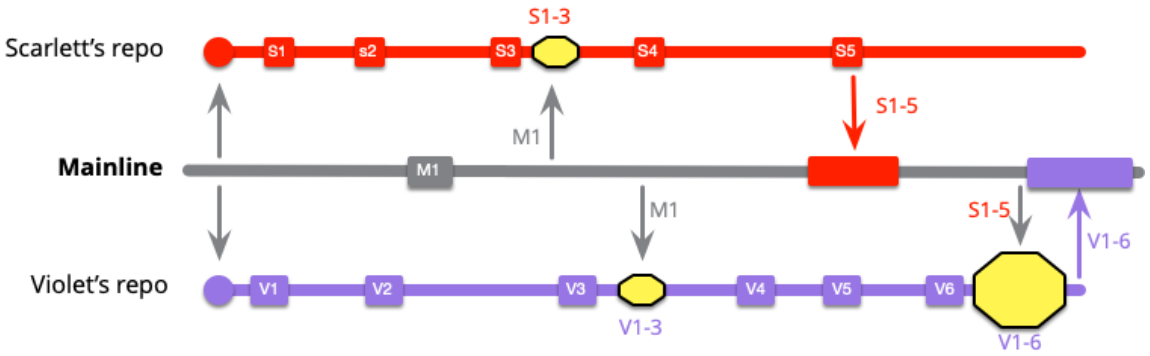
INTEGRATION PATTERNS

MAINLINE INTEGRATION



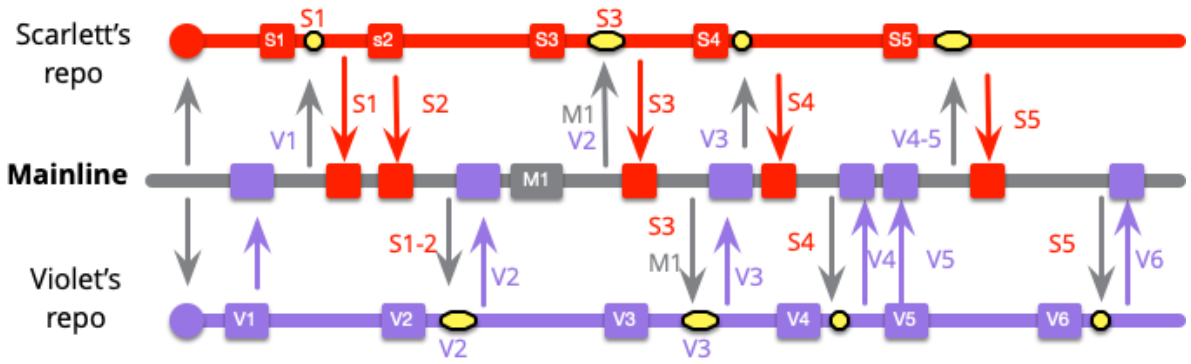
Integration Frequency

LOW FREQUENCY



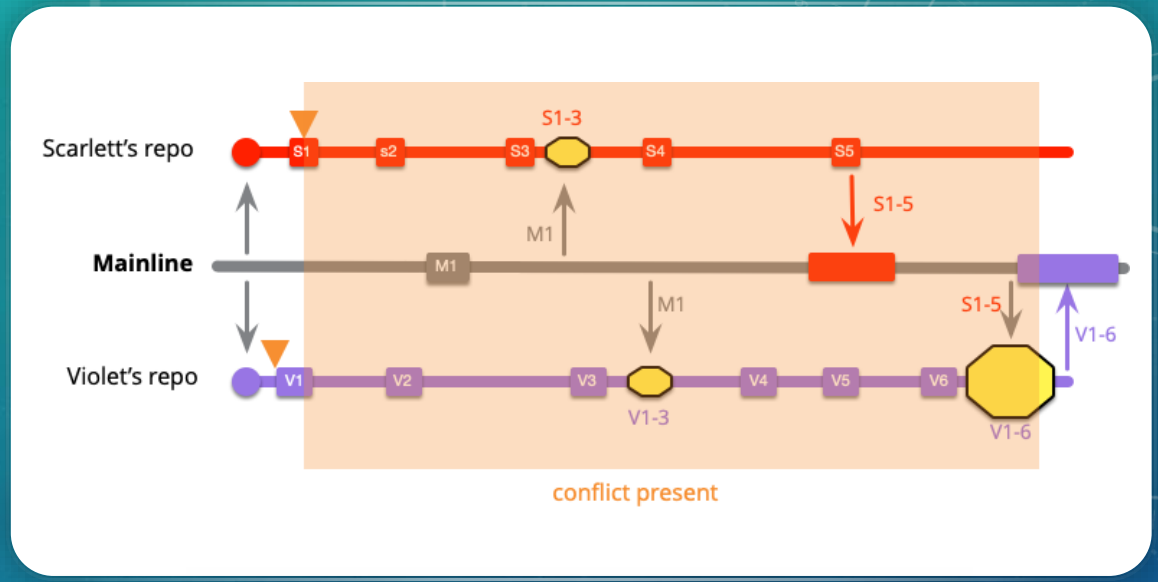
Integration Frequency

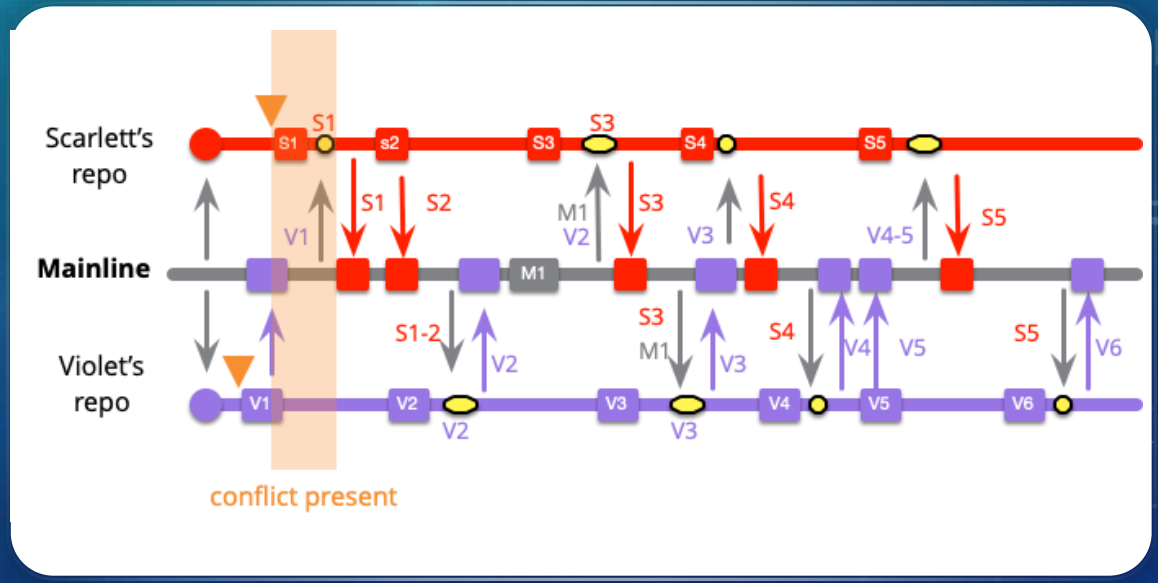
HIGH FREQUENCY



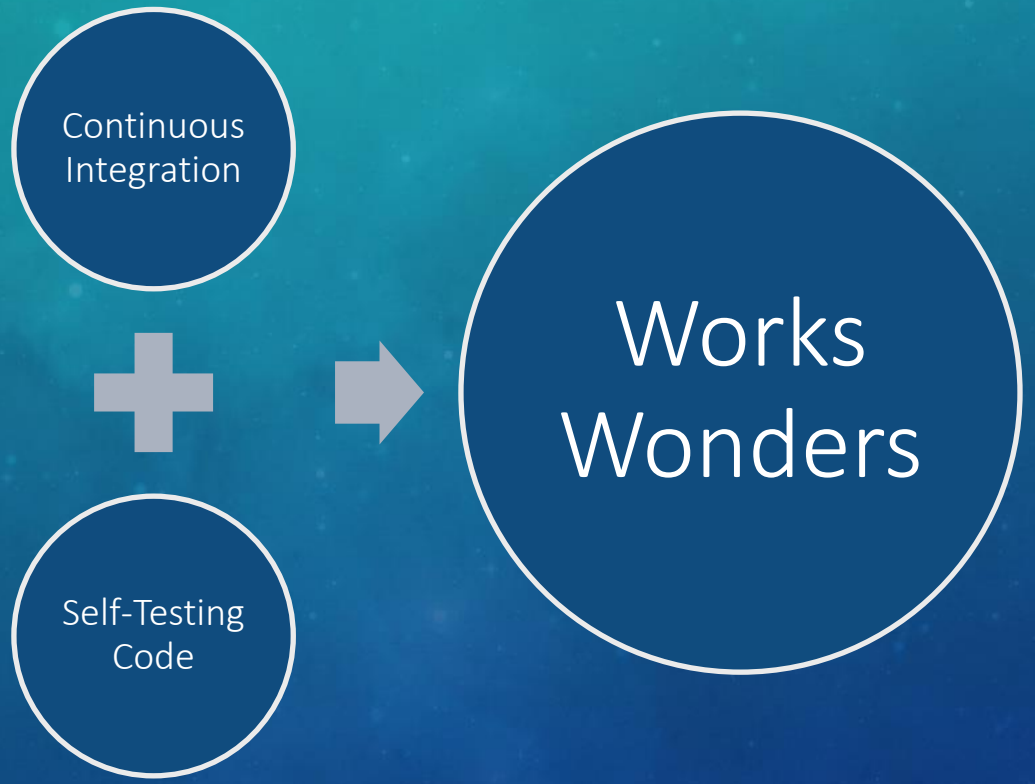
COMPARING FREQUENCIES

"if it hurts… do it more often"





CONTINUOUS INTEGRATION



COMPARING FEATURE BRANCHING AND CONTINUOUS INTEGRATION

If you know You can merge every day

Choose

Continuous Integration

If you do not know that

Choose

Feature Branching

COMPARING FEATURE BRANCHING AND CONTINUOUS INTEGRATION

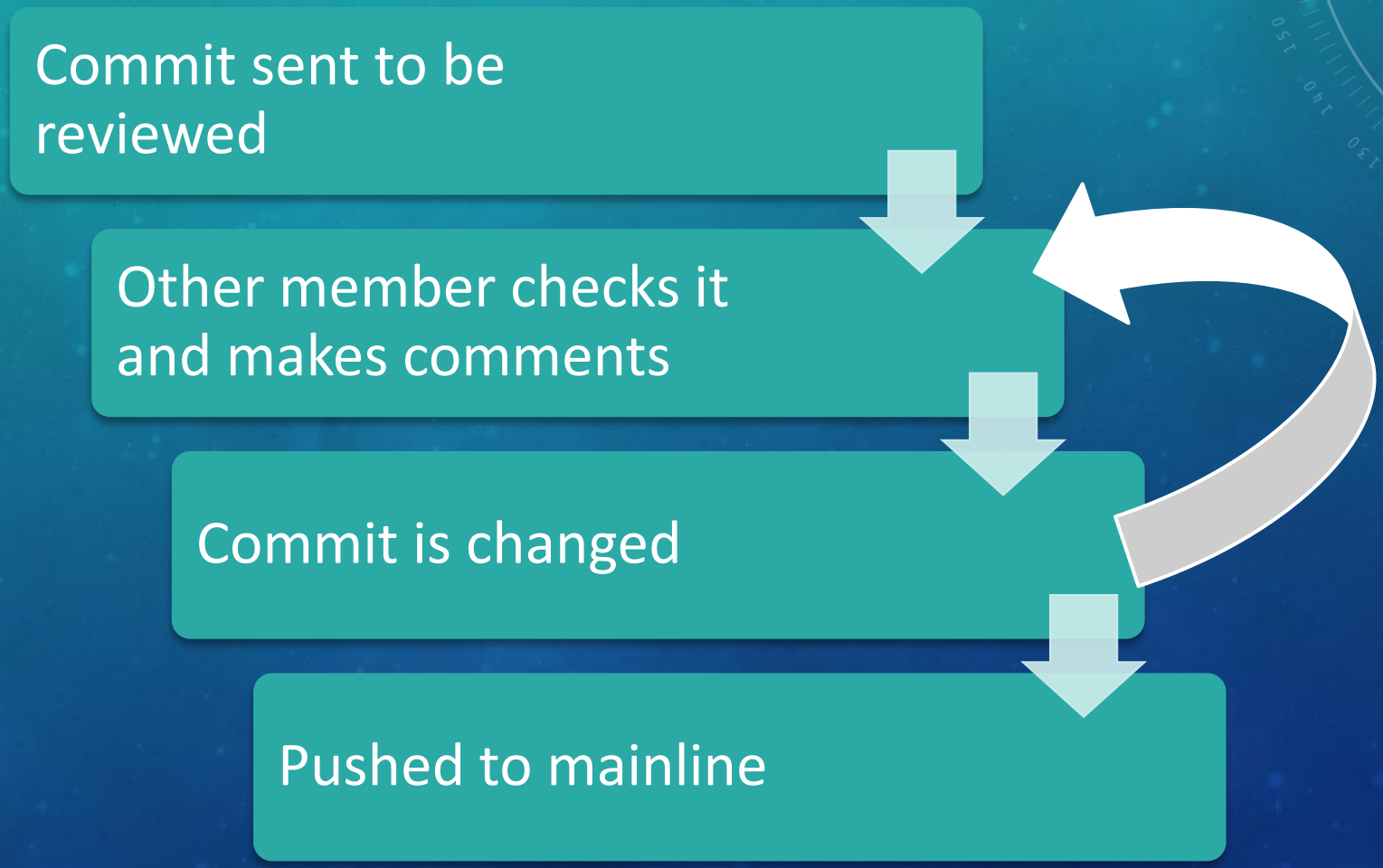
Feature Branching

- ✓ All the code in a feature can be assessed for quality as a unit
- ✓ Feature code only added to product when feature is complete
- ✗ Less frequent merges

Continuous Integration

- ✓ Supports higher frequency integration than feature length
- ✓ Reduced time to find conflicts
- ✓ Smaller merges
- ✓ Encourages refactoring
- ✗ Requires commitment to healthy branches (and thus self-testing code)
- ✓ Scientific evidence that it contributes to higher software delivery performance

PREINTEGRATION REVIEW

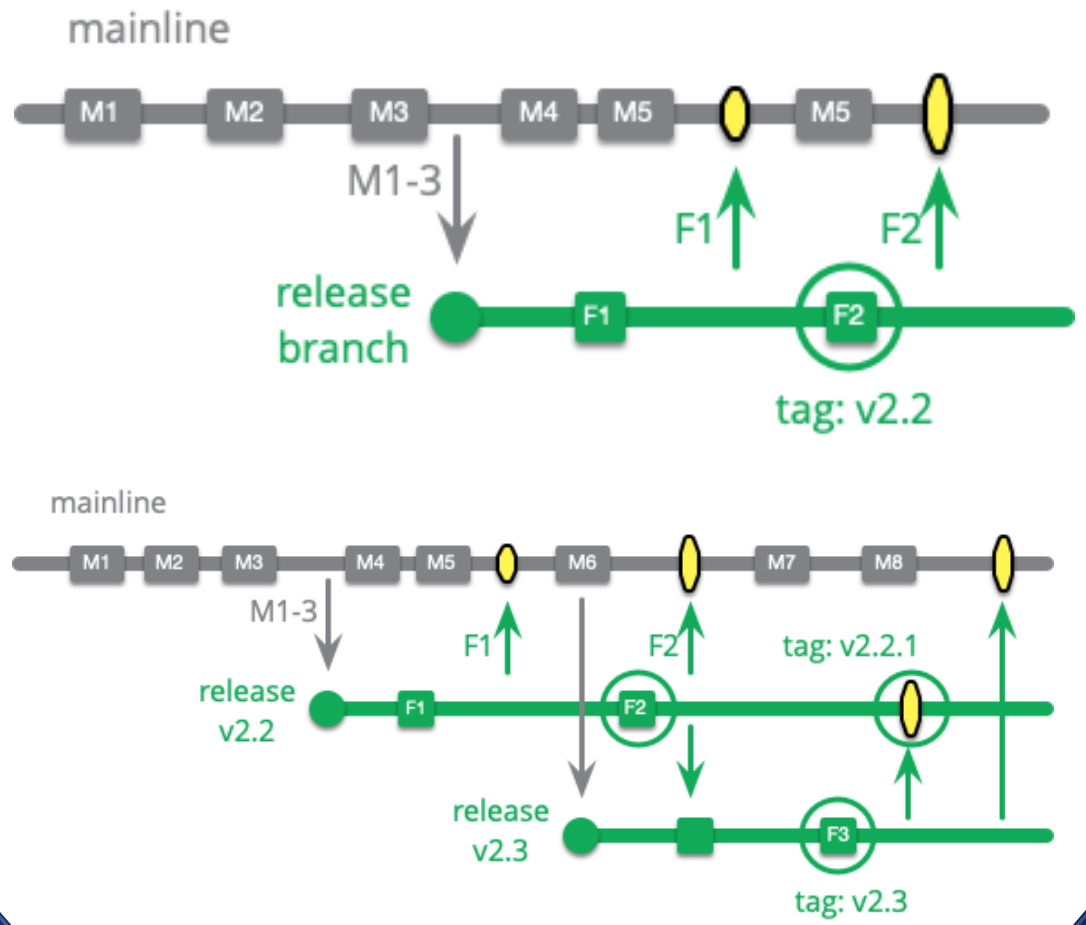


PREINTEGRATION REVIEW

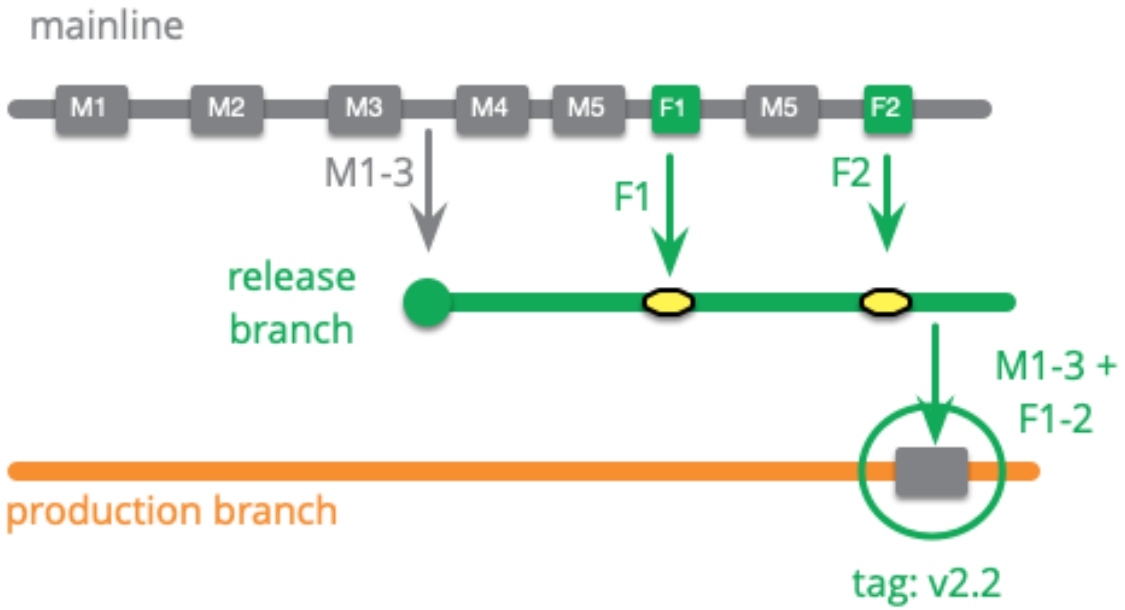
- Popular for Open Source projects
- Generates integration friction
- Modularity is very important for integration

MAINLINE TO PRODUCTION

RELEASE BRANCH

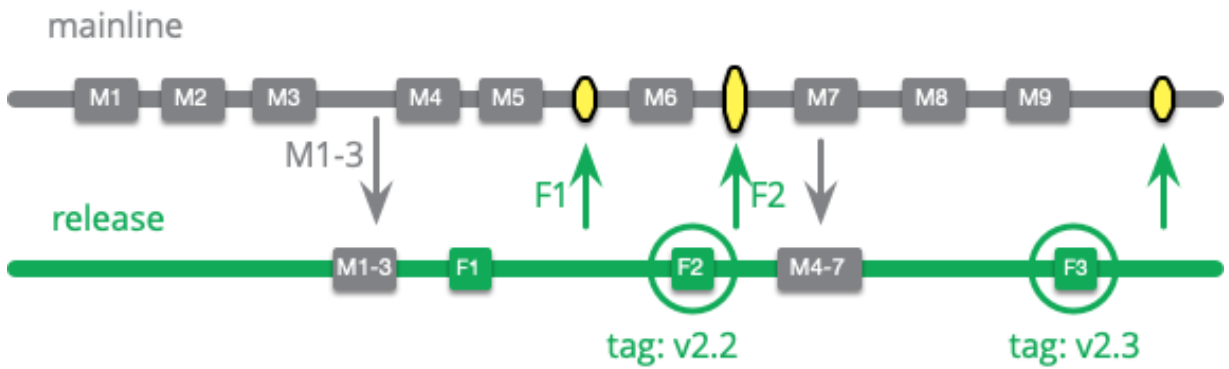


MATURITY BRANCH

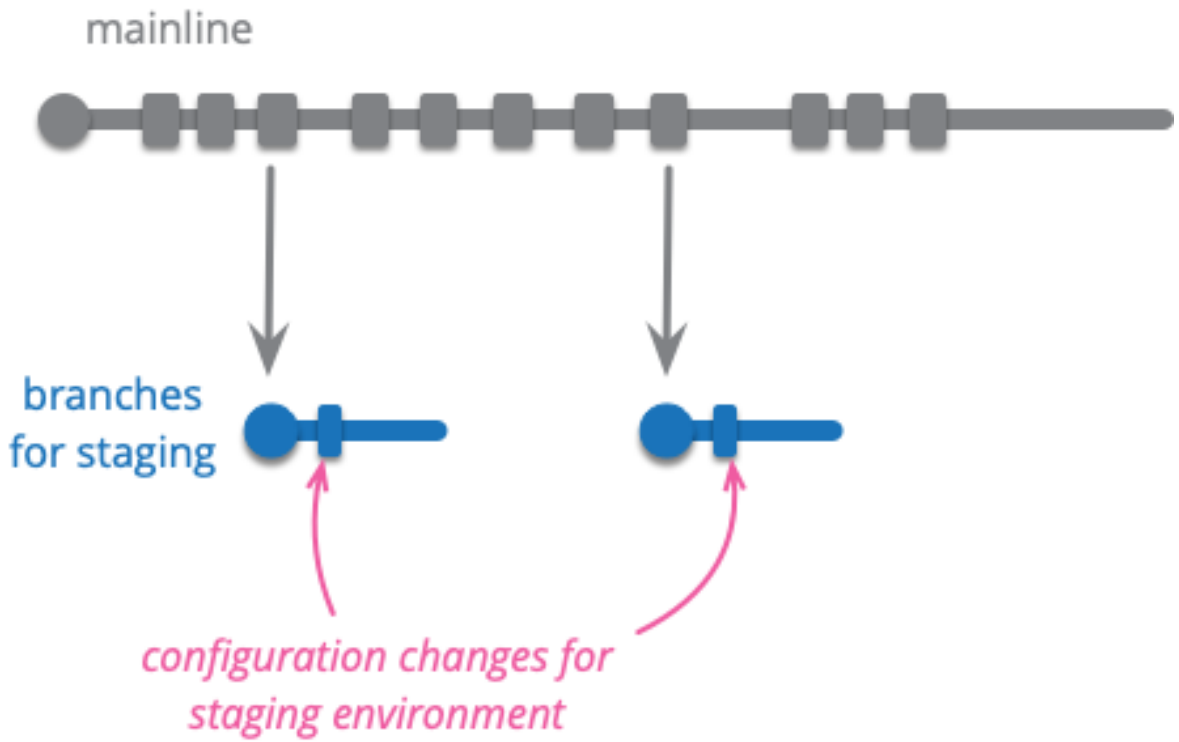


LONG LIVED RELEASE BRANCH

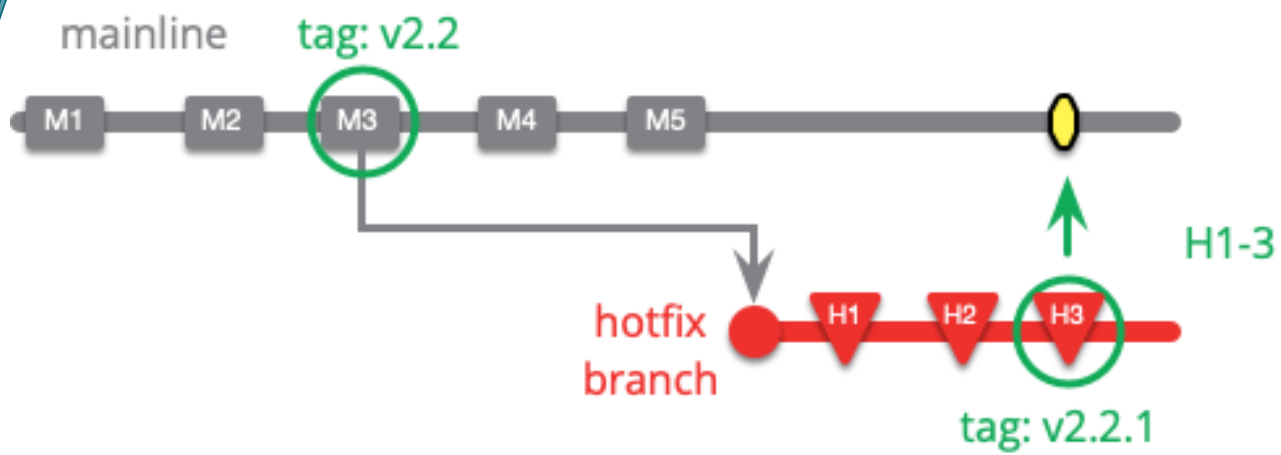
Release Branch + Maturity Branch



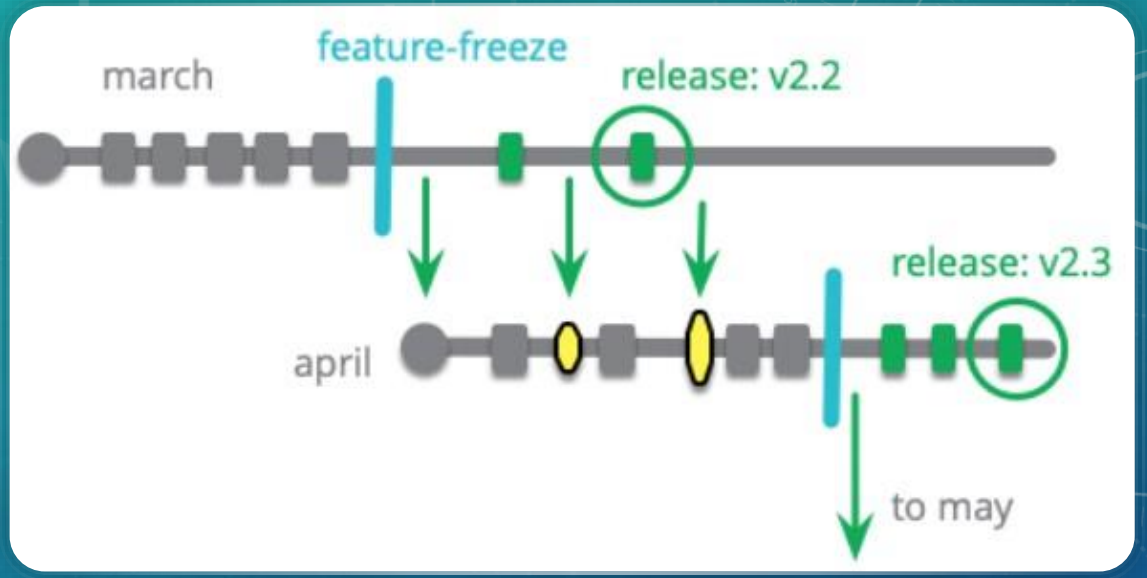
ENVIROMENT BRANCH

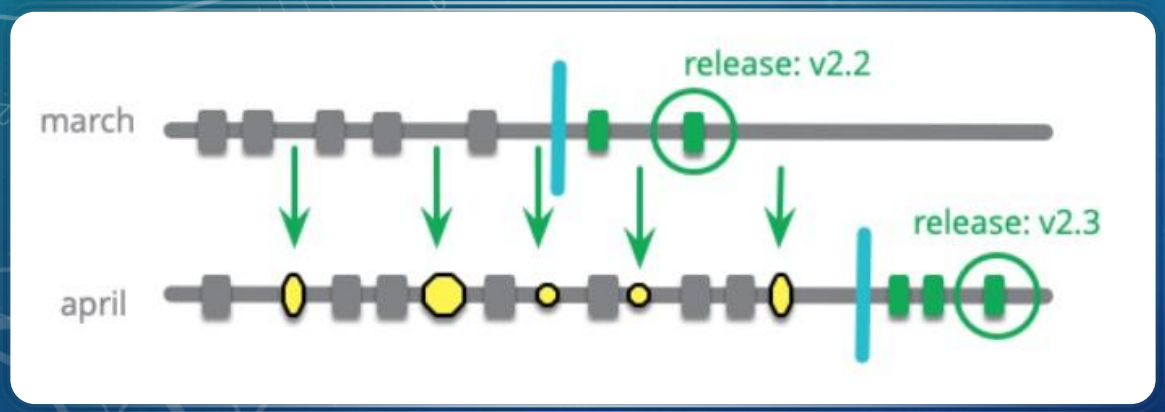


HOTFIX BRANCH



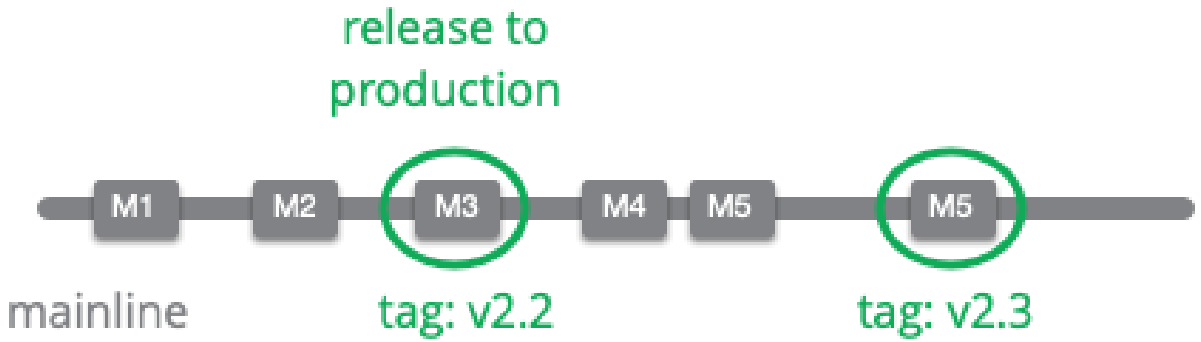
RELEASE TRAIN





Variation: Loading future trains

RELEASE-READY MAINLINE



OTHER PATTERNS

Experimental Branch

- Several experimental aproaches
- Code may be abandoned

Collaboration Branch

- Share code among collegues
- When code is not visible to others.

Team Integration Branch

- Used when work is divided into sub-groups
- MAINLINE of the sub-group