

_Feature Toggles

Karol Kreft

Szczecin PHP Meetup #9 - Experts invasion





FeatureBranch

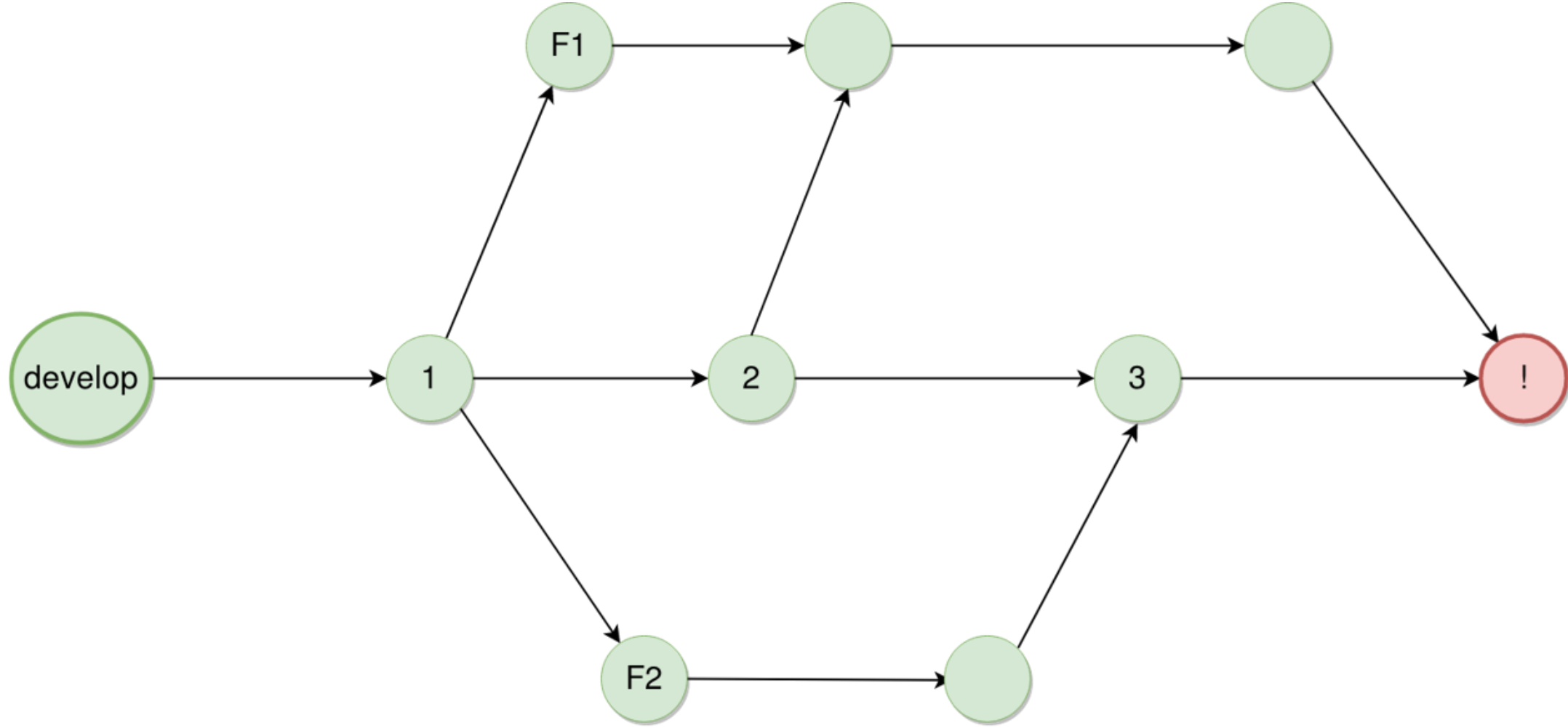


Advantages

- each developer can work on their own feature and be isolated from changes going on elsewhere
- it allows to choose the features for release by cherry-picking

Disadvantages

- Big Scary Merge (a.k.a. Merge Bomb)
- Semantic conflicts



Disadvantages



Steve Smith
@SteveSmithCD

Follow



Hi @InfoQ, your #GitFlow article is flat out wrong. It is 100% incompatible with #ContinuousIntegration and #ContinuousDelivery.
continuousdeliveryconsulting.com/blog/organisat ... / @danielbryantuk

Kenny Baas-Schwegler @kenny_baas

I won't go into the discussion about if gitflow is better or not then trunk based (it is not imo). But can people please stop calling using gitflow CI/CD. It is not continuous, you are building stuff in isolation.

11:59 PM - 9 Feb 2019

132 Retweets 283 Likes



Continuous Integration

” *Continuous Integration is a software development practice where members of a team **integrate** their work frequently.*

Martin Fowler, Chief Scientist, ThoughtWorks

” *Continuous Integration
doesn't get rid of bugs, but
it does make them
**dramatically easier to find
and remove.***

Martin Fowler, Chief Scientist, ThoughtWorks

Continuous Delivery

Continuous Delivery

” *Essentially, it is the practice of releasing **every good build** to users.*

*Jez Humble, co-author of Continuous Delivery: Reliable Software Releases Through Build, Test and Deployment Automation
CTO DevOps Research and Assessment*

Delivery
!=
Deployment

Delivery and Deployment

	Unit tests	Platform tests	Deliver to staging	Acceptance tests	Deploy on production	Post deploy tests
Delivery	AUTO	AUTO	AUTO	AUTO	MANUAL	AUTO
Deployment	AUTO	AUTO	AUTO	AUTO	AUTO	AUTO

Amazon deploys new software to production every 11.6 seconds.

*Amazon deploys new software to
production every 11.6 seconds.
In 2011.*

But what about...

- infrastructure?
- release planning schedule?
- informing users about new features?
- ?

But what about...

- infrastructure?
- release planning schedule?
- informing users about new features?
- ?

If you lack the process and quality discipline,
you will break production really fast.



Feature Toggles



Feature ~~Toggles~~

Flags, Bits, Flippers

” *“Feature Toggling” is a set of patterns which can help a team to deliver new functionality to users **rapidly but safely.***

Pete Hodgson, independent consultant, ThoughtWorks

Try to find meaningful context



Experimental algorithm

- Search algorithm
- Related products / posts
- New Payment gateway
- Login by Social Media
- Import / Export data feature

Permissions

- Access Control List
- Premium features
- Premium accounts

Features related to User

- New feature available only for some group of users.
 - Age
 - Gender
 - Most active users
 - Preferences

Geolocation

- Available only to specific region, country, city etc

A / B Tests

- More than one solution, which of them brings us more \$\$\$?
- Small but risk changes



I Am Devloper
@iamdevloper

Following



Which would you rather wear on a t-shirt?

Give me your answers now and get your obligated office Secret Santa present sorted.

51% No Deploy Fridays ✓

49% I Don't Deploy On Fridays

2,498 votes • 23 hours left

Canary release

- Only small subset of users will see the new feature
- If there are no critical bugs, extend a subset of users
- In case of some major errors occur, we are able to switch users back to the previous version of the software.

Before implementation

- How long each feature toggle will live in the system?
 - Unused feature toggle should be removed from the code.
 - Long-lived features may require a different configuration than the short-lived.

Implementation



```
public function search(phrase) {  
    // current implementation  
}
```

Implementation



```
private function newSearch(phrase) {  
    // new shiny implementation  
}  
  
private function oldSearch(phrase) {  
    // the old one  
}
```


Implementation



```
public function search(phrase) {  
    useNewAlgorithm = true;  
  
    if (useNewAlgorithm) {  
        return newSearch(phrase);  
    }  
  
    return oldSearch(phrase);  
}
```

Implementation



```
public function search(phrase) {  
    if (Feature::isEneabled("includePopularityInSearch") {  
        return newSearch(phrase);  
    }  
  
    return oldSearch(phrase);  
}
```

Implementation



```
[  
    'includePopularityInSearch' = true,  
    'displayRecomendationInSearch' = false  
    // ...  
]
```

Implementation



```
const INCLUDE_POPULARITY_IN_SEARCH = true;  
const DISPLAY_RECOMENDATION_IN_SEARCH = false;
```

Implementation

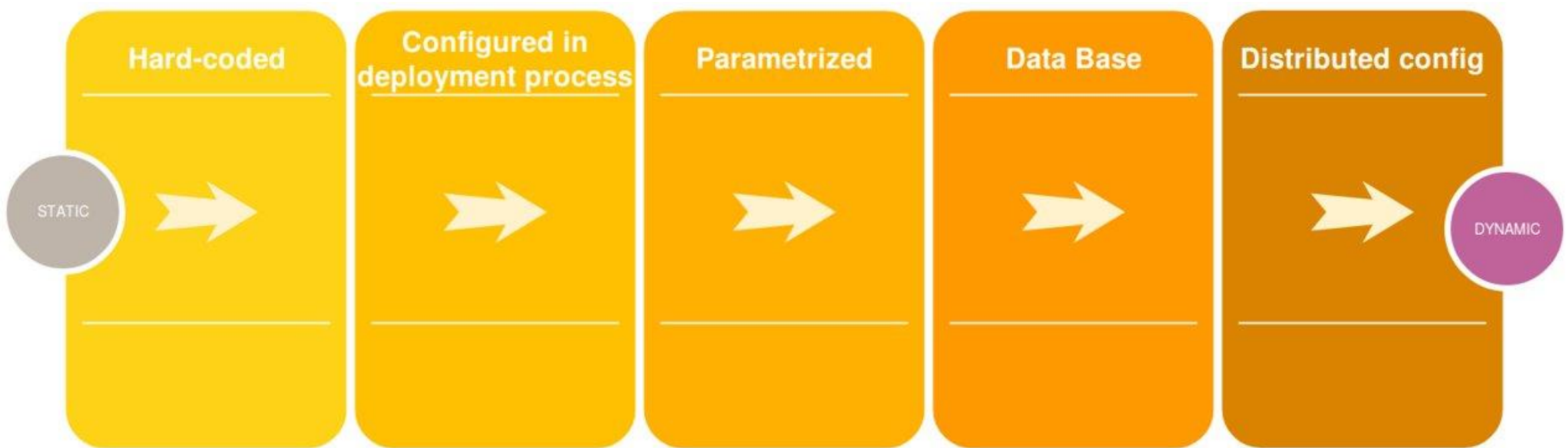


```
public function search(phrase) {  
    if (Feature::isEneabled("includePopularityInSearch", user) {  
        return newSearch(phrase);  
    }  
  
    return oldSearch(phrase);  
}
```

Ready packages

- `qandicate/toggle ^1.0`
- `dzunke/feature-flags-bundle:dev-master`
- `opensoft/rollout`

Configuration



Configuration



Apache
Zookeeper



LaunchDarkly

Costs

- Complexity, new abstraction to maintain
- Testing is harder, could require a method to administer current system state
- Technical debt, especially when lack of discussion and decision about removing unused feature toggles

Feature Toggles



Everywhere

Contact

Karol Kreft

Twitter - @karol_kreft

Web – kreft.net.pl



Sources

- [FeatureBranch, Martin Fowler](#)
- [Organisation Antipattern: Integration Feature Branching](#)
- [Continuous Integration, Martin Fowler](#)
- [Feature Toggles \(aka Feature Flags\), Pete Hodgson](#)
- [A journey into Feature Toggles, Pete Hodgson](#)
- [Velocity 2011: Jon Jenkins, Velocity Culture](#)
- [The Story of Apollo - Amazon's Deployment Engine, Werner Vogles](#)
- [Featureflags.io](#)
- [LanchDarkly - Feature Toggle Management For Continuous Delivery](#)

Photos

- [Photo by Diego PH on Unsplash](#)
- [Photo by Aaron Burden on Unsplash](#)
- Ned Stark from "Game of Thrones" TV Show.