

# DevOps and code management

Gunnar Rye Bergersen  
Associate professor, UiO

Oslo, Apr 26, 2022 – IN1030





Vi søker utviklere som sjefen  
absolutt ikke vil miste



**Mobil.** FINN har apps for både Android og iOS. Over 35% av all trafikk på FINN.no kommer via apps. Du jobber med utvikling enten på Android eller iOS, eller begge. Du kjenner til alle utfordringer det er å utvikle applikasjoner som må forholde seg til dårlig båndbredde og lite minne- og lagringskapasitet.



#### Vi vil at du søker, så send oss gjerne:

- Motivasjonen for å jobbe i FINN.no
- CV eller LinkedIn
- Karakterutskrift, dersom du er nyutdannet (<2 års erfaring)
- Har du noen prosjekter på Github, har du holdt presentasjoner eller er du er stolt av noe annet du har gjort, send oss gjerne en link til det.

Vi får mange åpne søknader, så det kan ta litt tid for du får tilbakemelding.

#### Ønsker du å vite mer om hvordan det er å være utvikler i FINN.no?

Vi har tro på å være veldig åpne i FINN om hva vi gjør og hvordan vi gjør det. Vi holder presentasjoner på konferanser som JavaZone, vi publiserer erfaringer på teknologibloggen vår, og vi deler åpen kildekode på Github.

#### FINN.no er også tilgjengelig i flere sosiale medier:

LinkedIn: <https://linkedin.com/company/finn.no>

Twitter: [https://twitter.com/finn\\_no](https://twitter.com/finn_no) [https://twitter.com/finn\\_tech](https://twitter.com/finn_tech)

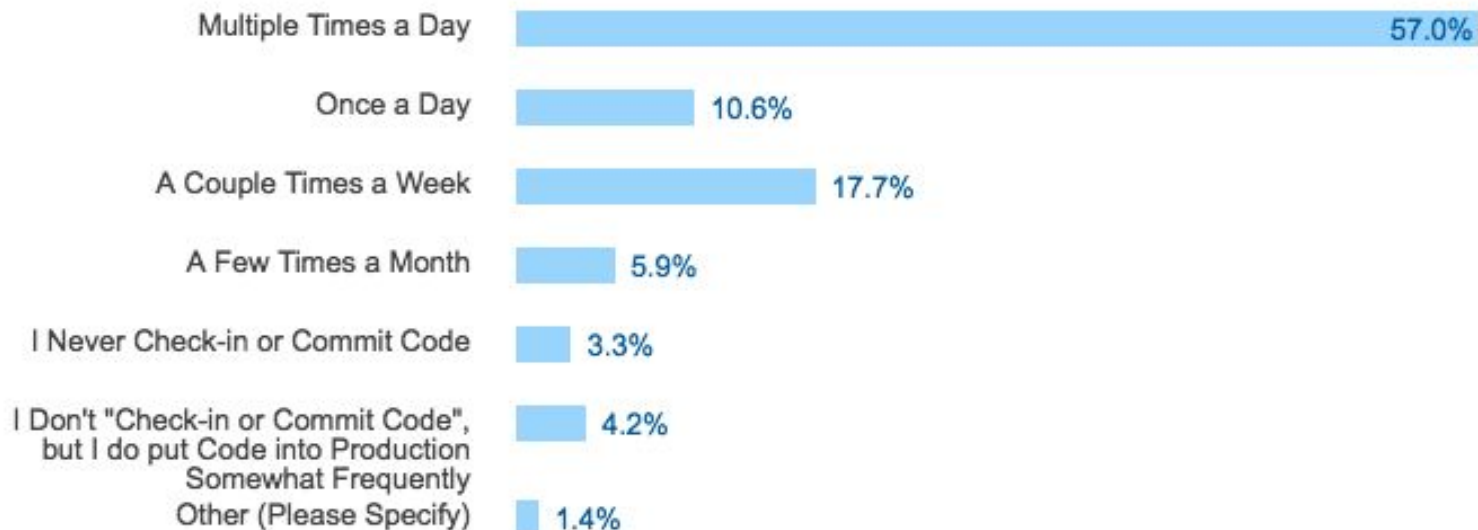
Facebook: <https://facebook.com/finn.no>

Instagram: [https://instagram.com/finn\\_no](https://instagram.com/finn_no)

Hver sommer har vi også tekniske sommerprosjekter i FINN. Til sommeren vil studentene jobbe sammen i team for å lage noe relevant og vårt mål er at funksjonaliteten som teamet utvikler i løpet av sommeren skal ut på FINN.no og brukes av tusenvis av brukere. Studentene vil få en reell opplevelse av hvordan det er å jobbe i FINN.

Følg med til høsten for ny søknadsrunde.

# Question: How often do you check in code?



46,599 responses

<http://stackoverflow.com/research/developer-survey-2016>

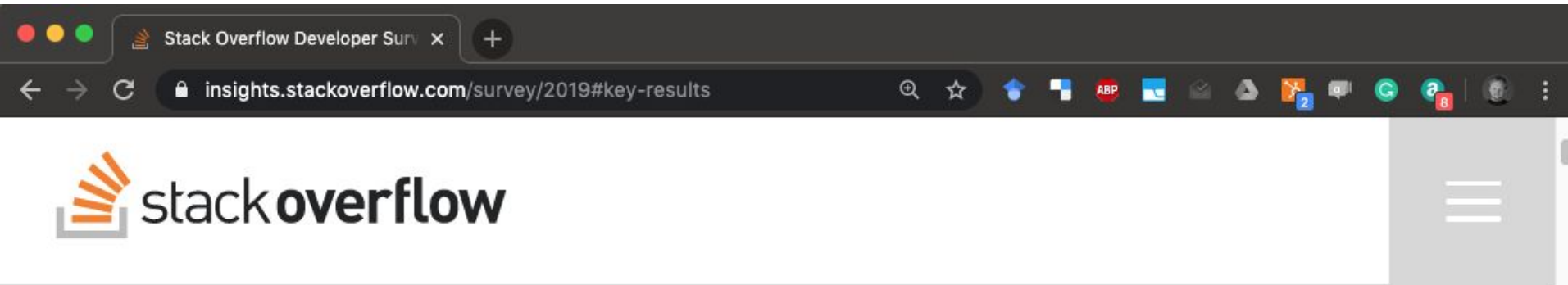
# Question: Are you a happy student?

## Developers Who Code are Happy Developers



*Percent of developers who love their job. 37,588 responses*

# StackOverflow 2019 highlights:



- **DevOps specialists** and site reliability engineers are among the highest paid, most experienced developers most satisfied with their jobs, and are looking for new jobs at the lowest levels.

# StackOverflow 2021



# So, what is DevOps?

## Definition [\[ edit \]](#)

---

Other than it being a cross-functional combination (and a [portmanteau](#)) of the terms and concepts for "development" and "operations", academics and practitioners have not developed a universal definition for the term "DevOps".<sup>[a][b][c][d]</sup> Most often, DevOps is characterized by key principles: shared ownership, workflow automation, and rapid feedback.

MANAGE

CREATE

PLAN

RELEASE

CONFIGURE

DEV

OPS

PACKAGE

MONITOR

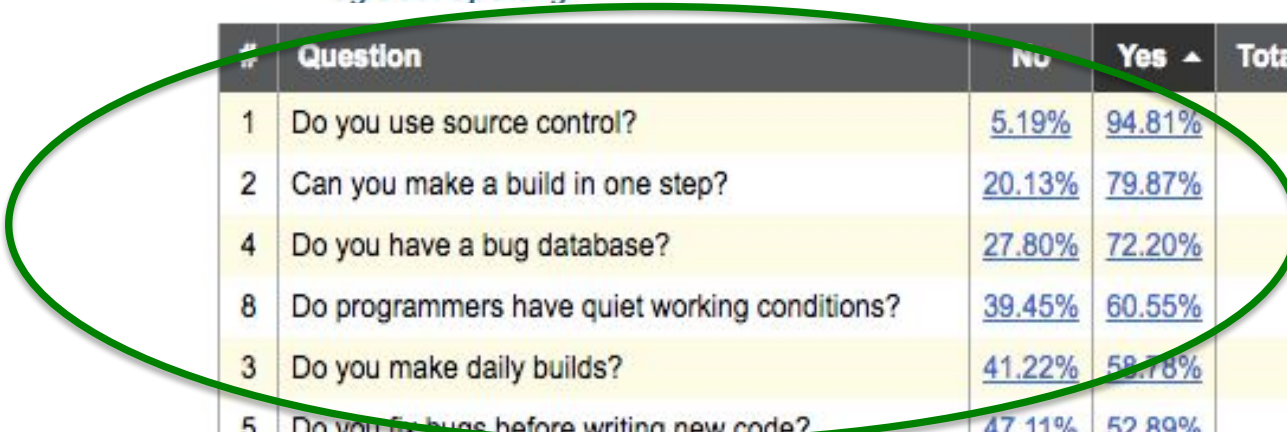
VERIFY

SECURE



# The Joel Test: 12 Steps to Better Code

by Joel Spolsky



#	Question	No	Yes	Total Responses
1	Do you use source control?	<a href="#">5.19%</a>	<a href="#">94.81%</a>	1,272
2	Can you make a build in one step?	<a href="#">20.13%</a>	<a href="#">79.87%</a>	1,267
4	Do you have a bug database?	<a href="#">27.80%</a>	<a href="#">72.20%</a>	1,266
8	Do programmers have quiet working conditions?	<a href="#">39.45%</a>	<a href="#">60.55%</a>	1,265
3	Do you make daily builds?	<a href="#">41.22%</a>	<a href="#">58.78%</a>	1,264
5	Do you fix bugs before writing new code?	<a href="#">47.11%</a>	<a href="#">52.89%</a>	1,265
10	Do you have testers?	<a href="#">49.25%</a>	<a href="#">50.75%</a>	1,263
6	Do you have an up-to-date schedule?	<a href="#">49.41%</a>	<a href="#">50.59%</a>	1,265
11	Do new candidates write code during their interview?	<a href="#">51.99%</a>	<a href="#">48.01%</a>	1,254
7	Do you have a spec?	<a href="#">54.95%</a>	<a href="#">45.05%</a>	1,263
9	Do you use the best tools money can buy?	<a href="#">59.27%</a>	<a href="#">40.73%</a>	1,262
12	Do you do hallway usability testing?	<a href="#">71.77%</a>	<a href="#">28.23%</a>	1,254

# Learning objectives

To understand the importance, principles, and tool support available for facilitating

- automation and
- team coordination

when developing, deploying and supporting code in (somewhat typical?) professional software development context.

Keywords: DevOps, code management (Git), containers (Docker), branching model

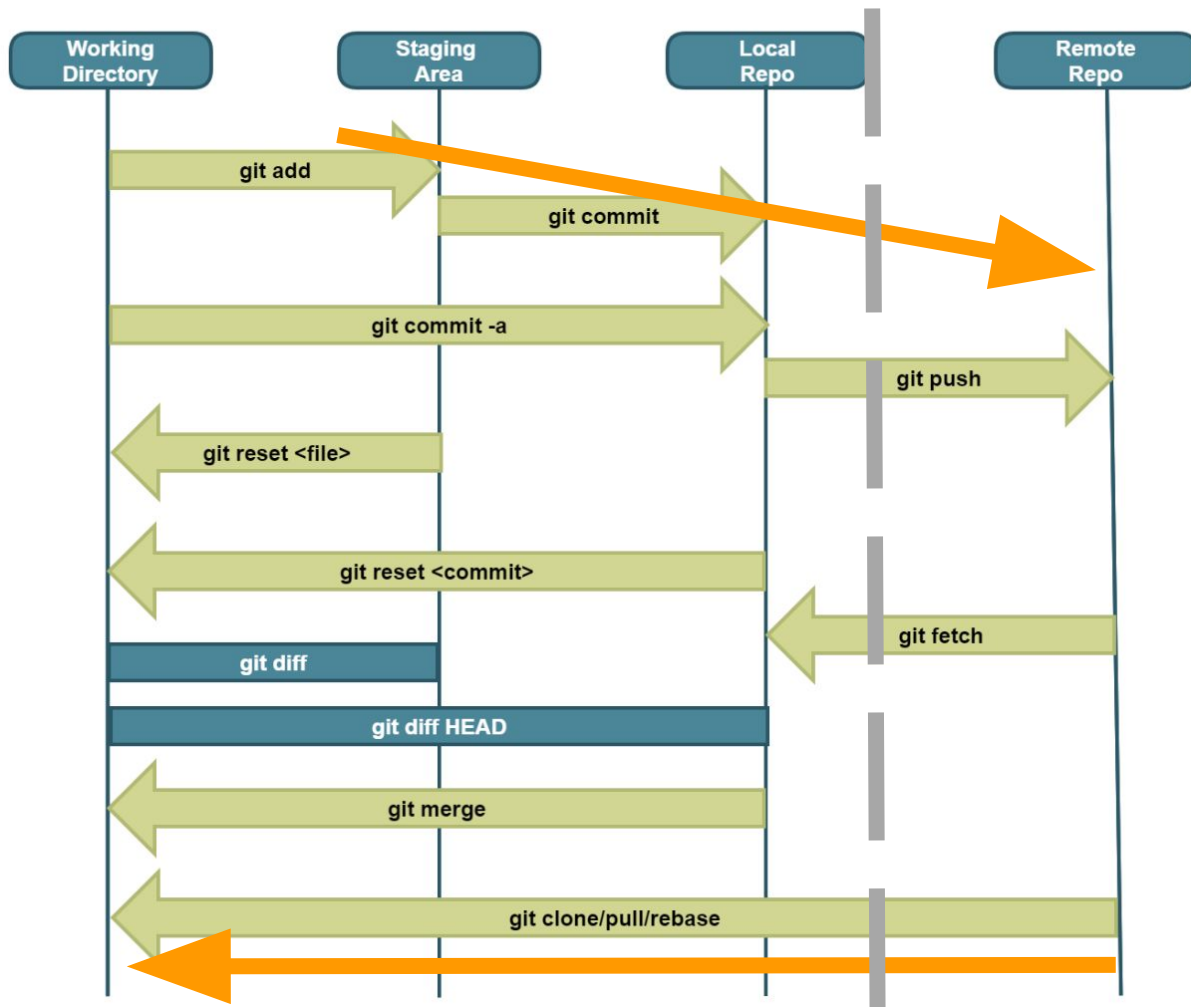
# Plan for lecture

- ~~1. Motivation and relevance of DevOps and code management (done)~~
2. Git: conceptual model and some live coding
3. DevOps automation for a concrete example
4. Coordination: Master / develop / feature branches
5. Summary

2 .Let's git going with some code ...

The git process involves four conceptual “areas” (3 local, 1 remote), separating you (left) from other devs (right)

Let’s try setting up a repo(sitory) using Git



Seen @IFI

Alternatively:

[https://www.reddit.com/r/ProgrammerHumor/comments/ub13j3/git\\_push\\_f\\_origin\\_master/](https://www.reddit.com/r/ProgrammerHumor/comments/ub13j3/git_push_f_origin_master/)

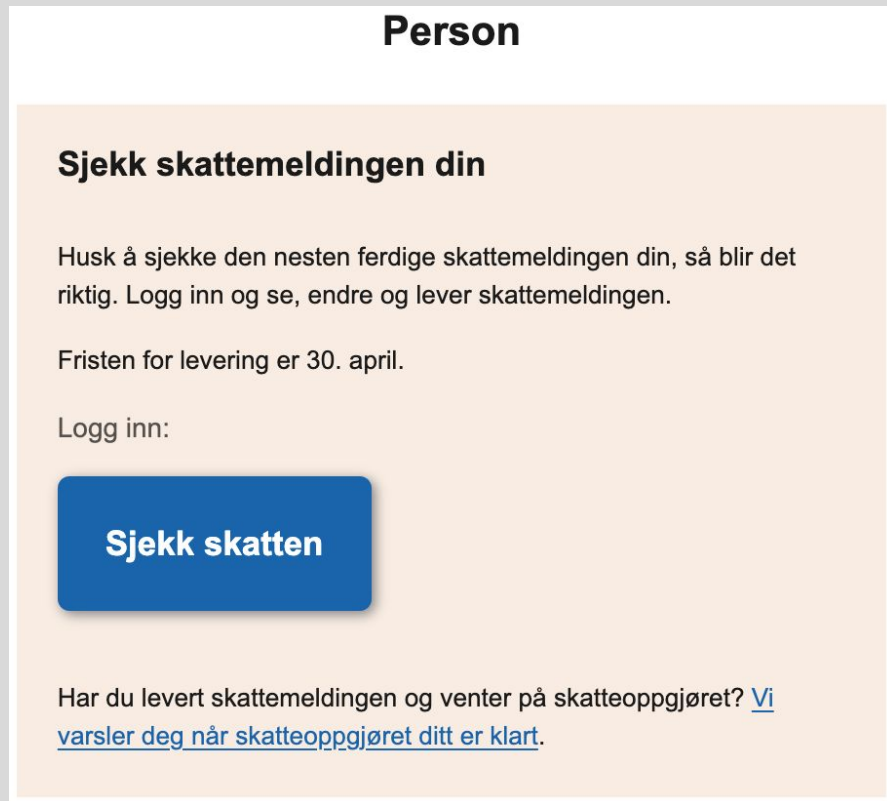


### 3. DevOps automation

# Let's think about what (DevOps) automation is required to calculate income tax at the Skatteetaten webpage

- Focus on end to end
- Containers
- Continuous integration/delivery/deployment

When tired, let's do an exercise (I'll explain its purpose afterwards)



**Person**

### Sjekk skattemeldingen din

Husk å sjekke den nesten ferdige skattemeldingen din, så blir det riktig. Logg inn og se, endre og lever skattemeldingen.

Fristen for levering er 30. april.

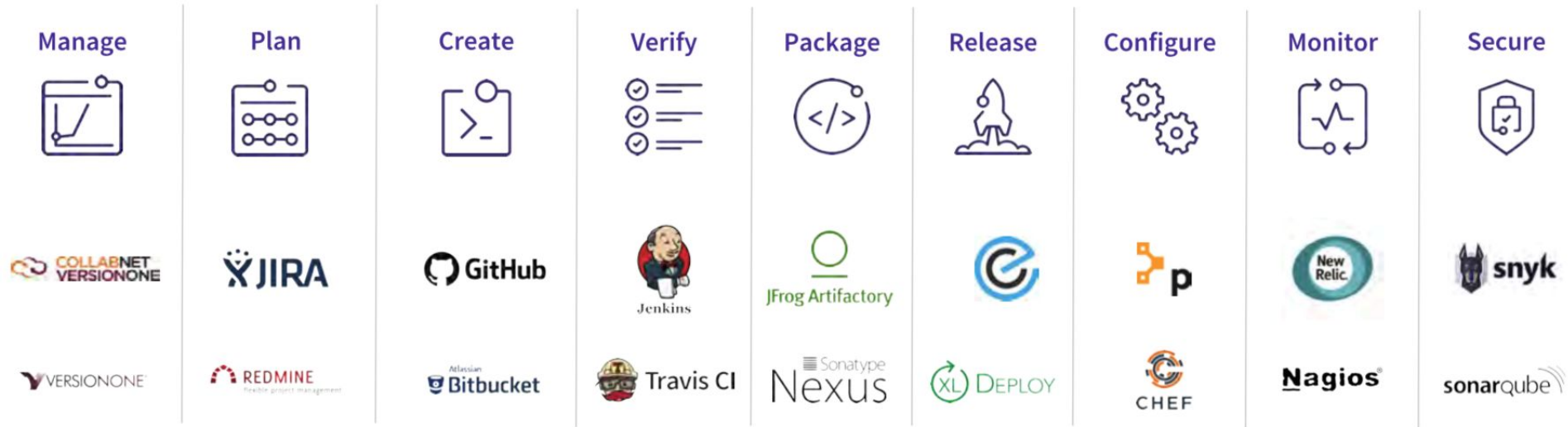
Logg inn:

**Sjekk skatten**

Har du levert skattemeldingen og venter på skatteoppgjøret? [Vi varsler deg når skatteoppgjøret ditt er klart.](#)



# Today's toolchain limits a faster DevOps lifecycle



Tool support is an integral part of automation



# Automate what can be automated



## EMERGE FROM BUILD HELL

Emerge from Build Hell today through Gradle, the modern open source polyglot build automation system. Automate and integrate your DevOps toolchain with a concise and expressive build programming language. End long build times. End code freezes. End build script chaos. End deathmarches. End bug regressions. End broken release processes.

# Automation (cont).



# Jenkins

Build great things at any scale

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

[Download Jenkins](#)

- People
- Build History
- Project Relationship
- Check File Fingerprint
- My Views

**Build Queue**

No builds in the queue.


**Build Executor Status**


- 1 Idle
- 2 **STAMP-Develop** #29

- All
- Develop Branch
- Master Branch

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">WebPages-Master</a>	7 days 13 hr - #14	1 mo 23 days - #2	28 sec
		<a href="#">WebPages</a>	7 days 13 hr - #60	1 mo 11 days - #36	14 sec
		<a href="#">TestAdmin-Master</a>	5 mo 16 days - #21	9 mo 10 days - #15	3 min 2 sec
		<a href="#">TestAdmin-Develop</a>	1 mo 24 days - #212	2 mo 1 day - #208	1 min 11 sec
		<a href="#">STAMP-Master</a>	N/A	26 days - #1	3.2 sec
		<a href="#">STAMP-Develop</a>	5 days 8 hr - #28	2 mo 7 days - #6	16 sec
		<a href="#">Scripts</a>	1 mo 5 days - #146	11 mo - #102	7.3 sec
		<a href="#">Score-Master</a>	5 mo 19 days - #3	N/A	58 sec
		<a href="#">Score-Develop</a>	8 hr 3 min - #66	5 mo 1 day - #47	48 sec
		<a href="#">NETS-Develop</a>	1 yr 7 mo - #61	2 yr 5 mo - #27	23 sec

Example of Jenkins building a deployment of the develop branch for our test server. We don't push to production branch manually

-  People
-  Build History
-  Project Relationship
-  Check File Fingerprint
-  My Views

**Build Queue** 

No builds in the queue.

**Build Executor Status** 

- 1 Idle
- 2 Idle

All Develop Branch Master Branch

S	W	Name	Last Success ↓	Last Failure	Last Duration
		<a href="#">STAMP-Develop</a>	34 sec - <a href="#">#29</a>	2 mo 7 days - <a href="#">#6</a>	21 sec
		<a href="#">Score-Develop</a>	8 hr 4 min - <a href="#">#66</a>	5 mo 1 day - <a href="#">#47</a>	48 sec
		<a href="#">GradleScripts</a>	5 days 6 hr - <a href="#">#48</a>	1 mo 0 days - <a href="#">#44</a>	14 sec
		<a href="#">GradlePlugins</a>	6 days 10 hr - <a href="#">#71</a>	29 days - <a href="#">#65</a>	48 sec
		<a href="#">WebPages-Master</a>	7 days 13 hr - <a href="#">#14</a>	1 mo 23 days - <a href="#">#2</a>	28 sec
		<a href="#">WebPages</a>	7 days 13 hr - <a href="#">#60</a>	1 mo 11 days - <a href="#">#36</a>	14 sec
		<a href="#">Scripts</a>	1 mo 5 days - <a href="#">#146</a>	11 mo - <a href="#">#102</a>	7.3 sec
		<a href="#">Deployment</a>	1 mo 6 days - <a href="#">#321</a>	8 hr 3 min - <a href="#">#323</a>	1 min 17 sec

5. Coordination: master / develop / feature branches

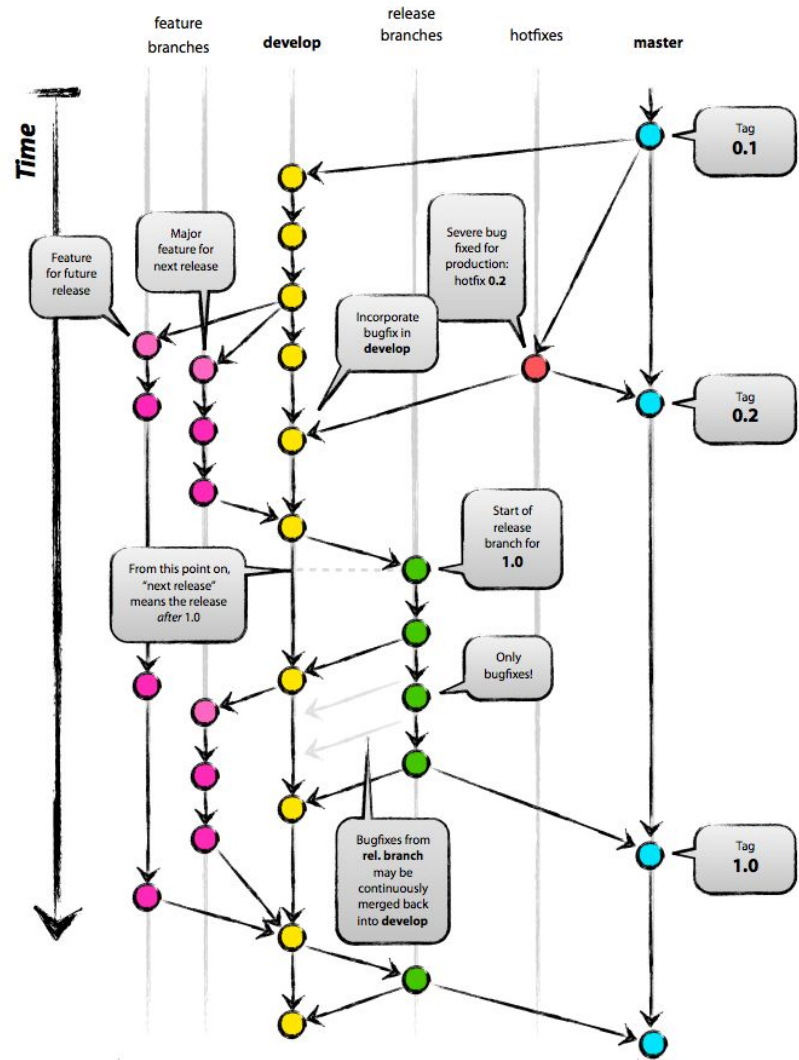
# Branches: a typical git production environment setup

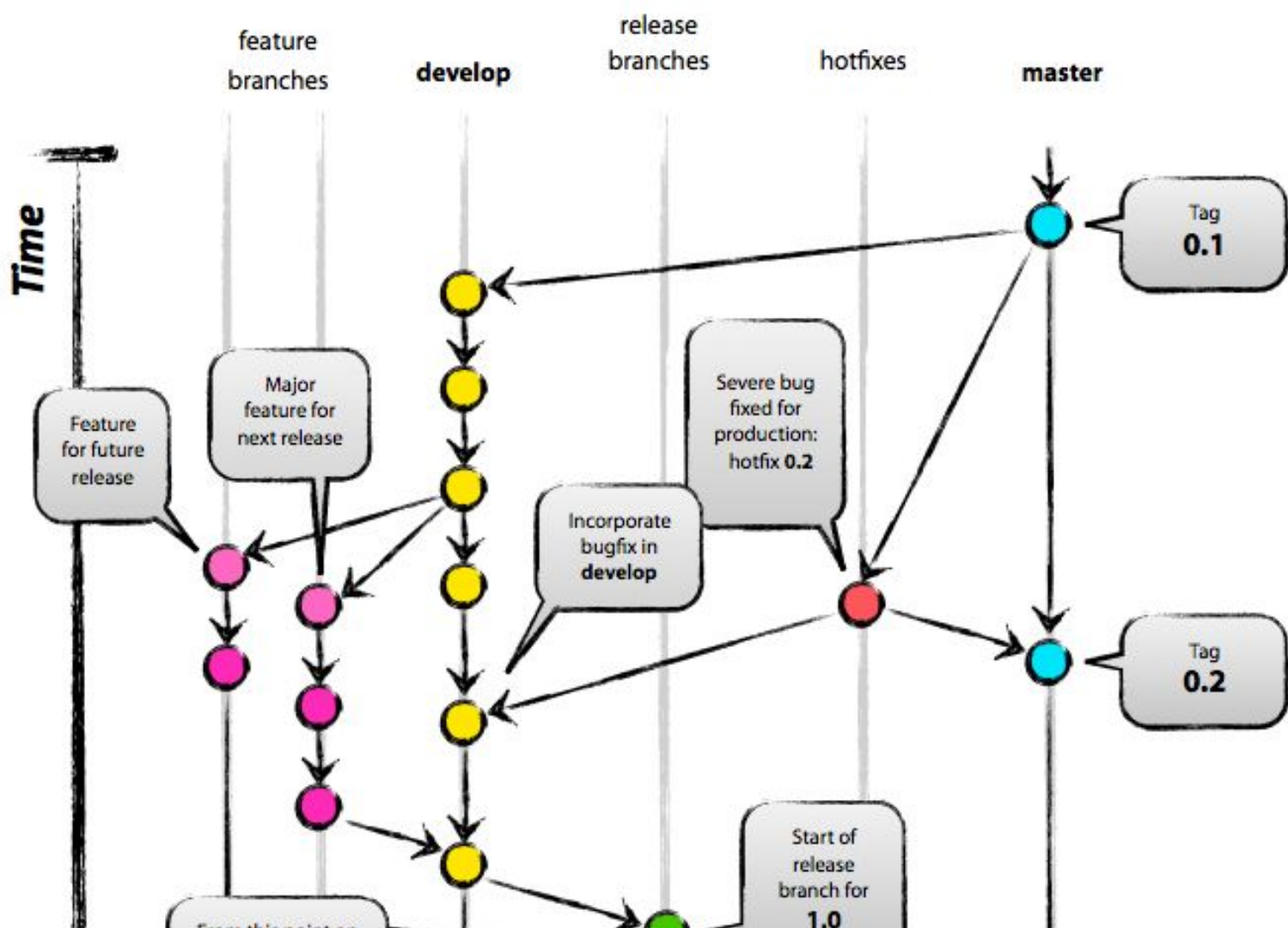
Note: I'm not happy about how Sommerville presents this in Figure 10.7. But note that practices can vary and the setup shown here is (somewhat?) typical of a small-medium sized SaaS team the last 5+ years.

Author: Vincent Driessen

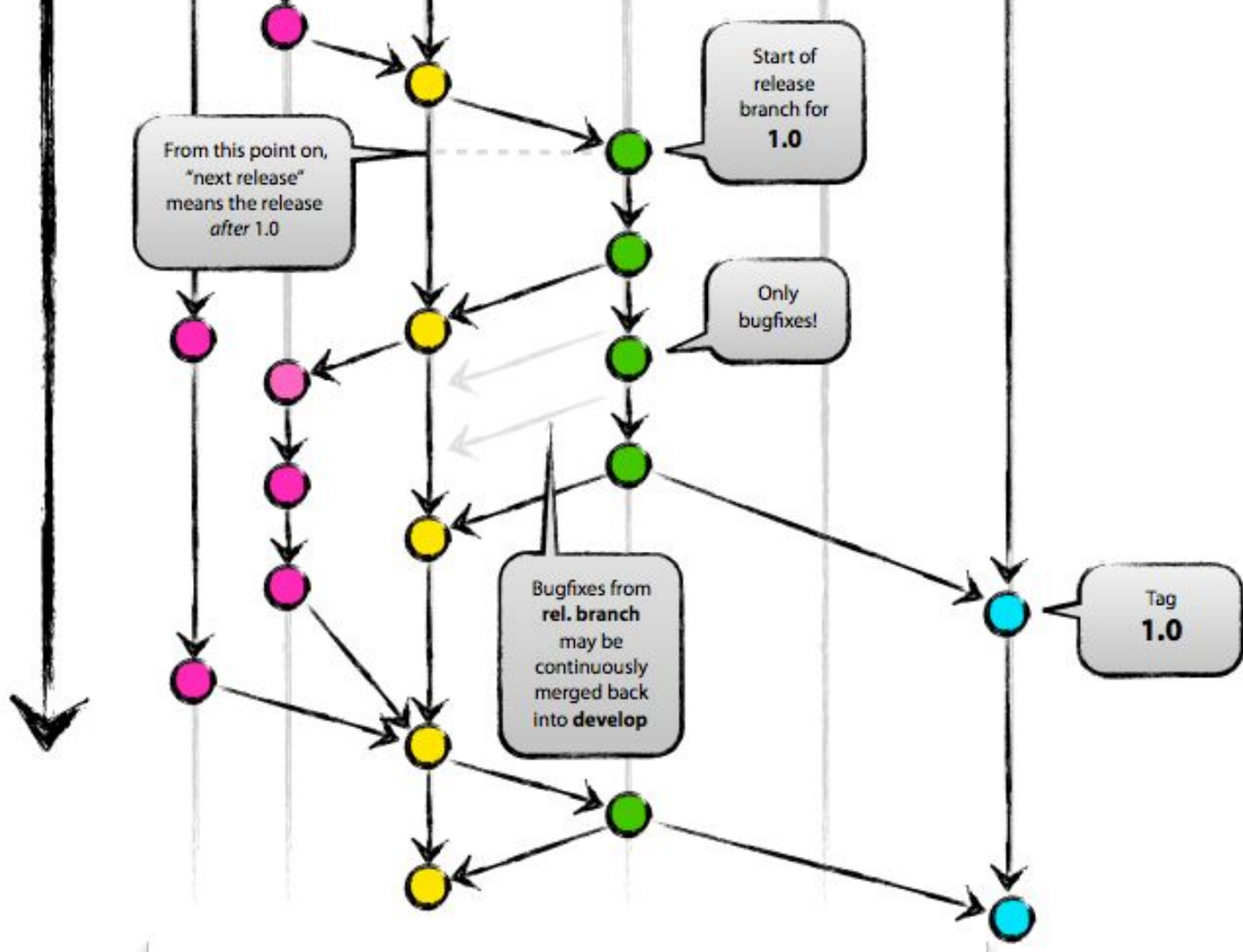
Original blog post: <http://nvie.com/posts/a-successful-git-branching-model>

License: Creative Commons BY-SA










## 5. Summary

# Teams cannot work without tools and best practices (DevOps) to ship working code to end users at scale



#	Question	No	Yes ^	Total Responses
1	Do you use source control?	<a href="#">5.19%</a>	<a href="#">94.81%</a>	1,272
2	Can you make a build in one step?	<a href="#">20.13%</a>	<a href="#">79.87%</a>	1,267
4	Do you have a bug database?	<a href="#">27.80%</a>	<a href="#">72.20%</a>	1,266
8	Do programmers have quiet working conditions?	<a href="#">39.45%</a>	<a href="#">60.55%</a>	1,265
3	Do you make daily builds?	<a href="#">41.22%</a>	<a href="#">58.78%</a>	1,264
5	Do you fix bugs before writing new code?	<a href="#">47.11%</a>	<a href="#">52.89%</a>	1,265
10	Do you have testers?	<a href="#">49.25%</a>	<a href="#">50.75%</a>	1,263
6	Do you have an up-to-date schedule?	<a href="#">49.41%</a>	<a href="#">50.59%</a>	1,265

Tips for :

- Start using Git today by registering an account on Github [github.com](https://github.com)
- Use the Github desktop (GUI) initially
- Make mistakes (lots of them)