Software development, procurement, & management fundamentals

Software development practices

Part 4 of 5

Presented by 18F for:
Office of Child Care, HHS

August, 2022

Software development, procurement, & management fundamentals series

1 Agile management

2 Product ownership

3 User-centered design

4 Software development practices

5 Agile Contracting

What is 18F?

18F is a technology and design consultancy for the U.S. Government, inside the government.

We share the same motivations as you: **delivering great service to the public.**

Greg Walker

Experience

- US Army Corps of Engineers, Research and Development Center: Hurricane and flooding disaster response and recovery
- 18F: State child welfare IT modernization (sorta!)
  - Lots of acquisition
- 18F: State Medicaid IT modernization
  - Lots of acquisition
- 18F: Federal Medicaid IT oversight policy
  - How to make acquisition easier and more successful for state/tribal/territorial partners

4/ Software development practices

18F

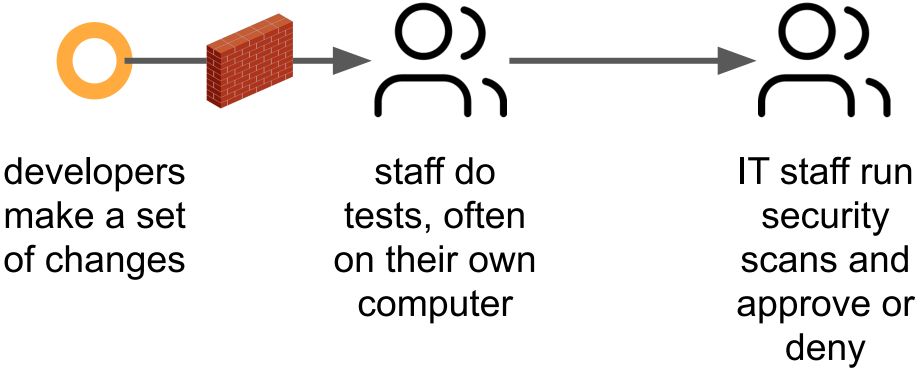
Breaking down silos and enabling continuous, iterative delivery

DevSecOps combines software development, security, and operations into one team, and automates as many of their processes as possible. People focus on the hard problems and let the computers handle the repetitive stuff.

A traditional delivery process

developers
make a set
of changes

A traditional delivery process

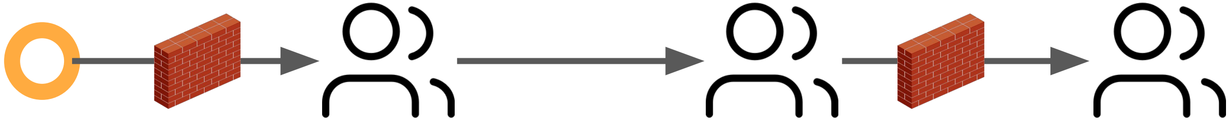


developers make a set of changes

staff do tests, often on their own computer

IT staff run security scans and approve or deny

A traditional delivery process



developers make a set of changes

staff do tests, often on their own computer

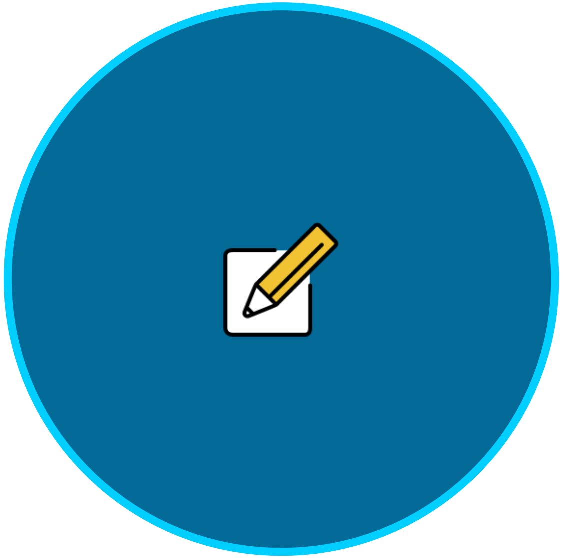
IT staff run security scans and approve or deny

product technical staff deploy the product

```
Processes: 123 total, 3 running, 120 sleeping, 558 threads                    08:36:09
Load Avg: 1.75, 1.53, 1.49  CPU usage: 25.86% user, 18.93% sys, 55.62% idle
SharedLibs: 3908K resident, 5760K data, 0B linkedit.
MemRegions: 44714 total, 3368M resident, 77M private, 1118M shared.
PhysMem: 921M wired, 5200M active, 758M inactive, 6948M used, 1243M free.
VM: 238G vsize, 1034M framework vsize, 4797528(2) pageins, 0(0) pageouts.
Networks: packets: 581628/454M in, 462610/68M out.
Disks: 229509/3409M read, 418661/7924M written.

PID   COMMAND      %CPU TIME     #TH  #WQ  #POR #MREG RPRVT  RSHRD  RSIZE
1477  top          12.9 00:01.38 1/1  0    24   33    1420K+ 244K   1998K+
1466- cvmsComp_i38 0.0  00:00.04 1    0    18   36    1116K  9528K  5760K
1463  bash         0.0  00:00.00 1    0    17   25    296K   856K   968K
1462  login        0.0  00:00.01 1    0    22   62    616K   3200K  2448K
1459  cvmsComp_x86 0.0  00:00.03 1    0    18   34    1592K  9528K  6220K
1456- Cathode      8077 00:10.88 5    2    127  267   20M+   92M+   65M+
1454  launchd      0.0  00:00.00 2    0    37   46    236K   428K   660K
1452  quicklookd   0.0  00:00.48 6    2    88-  155   21M-   17M    58M-
1451  ocspd        0.0  00:00.01 2    0    42   40    736K   3192K  2152K
1450  mdworker     0.0  00:00.06 3    1    48   67    1636K  16M    4284K
1294- Google Chrom 0.3  00:42.07 4    1    93   778   48M    89M    80M
1267- DashboardCli 0.0  00:01.27 5    2    128  228   14M    26M    21M
1266  DashboardCli 0.0  00:02.39 5    2    129  330   40M    43M    97M
1192- Google Chrom 0.8  00:10.10 4    1    93   348   19M-   87M    43M-
1014  dd           0.0  00:00.00 1    0    14   23    180K   240K   436K
```

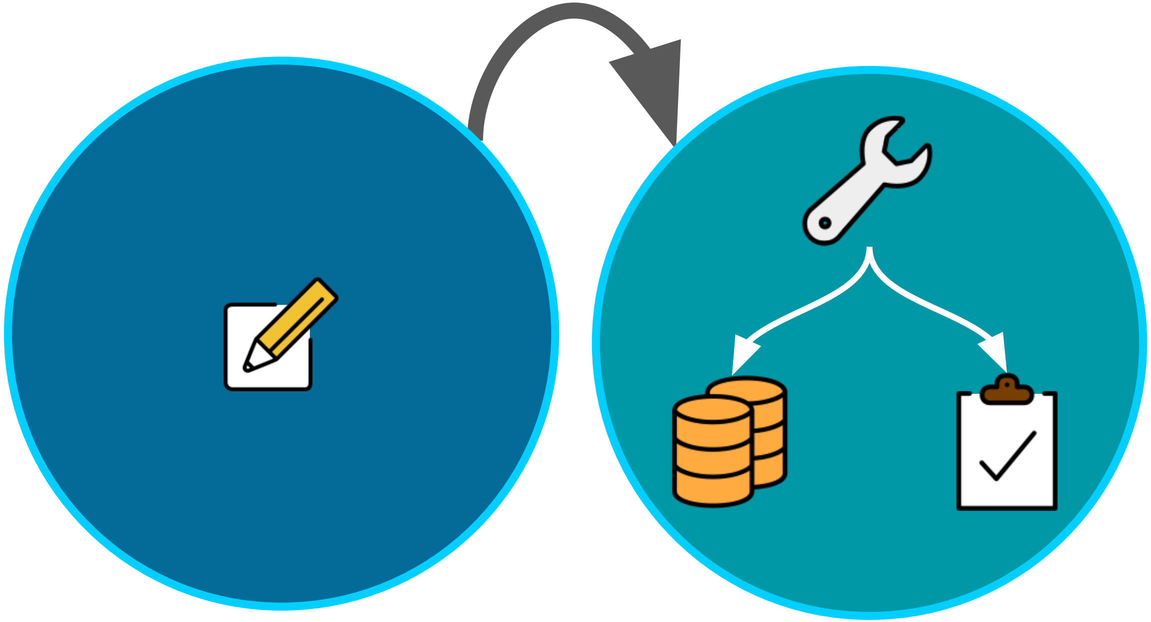
Continuous delivery



Something changes...

- Code changes
- Updates to text

Continuous delivery

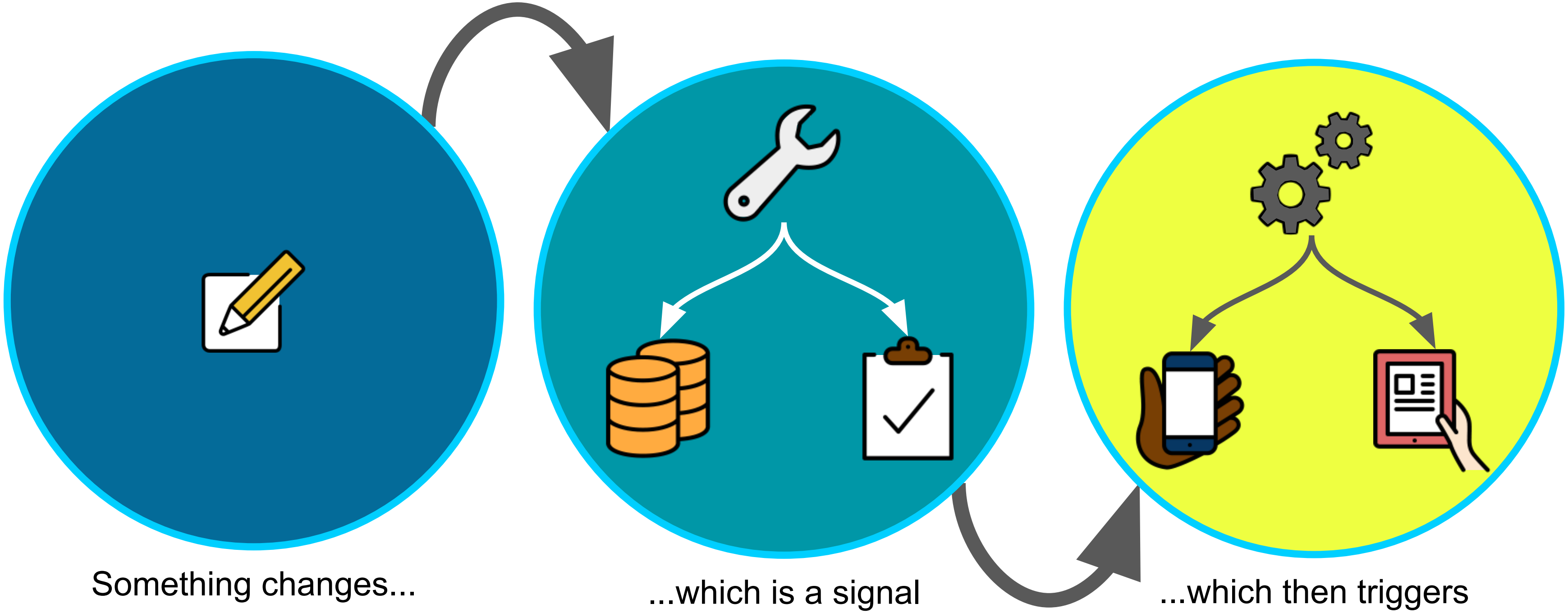


Something changes...

...which is a signal
for automated testing
to start...

- Functional testing of business logic
- Accessibility testing
- Code quality testing
- Security scanning

Continuous delivery

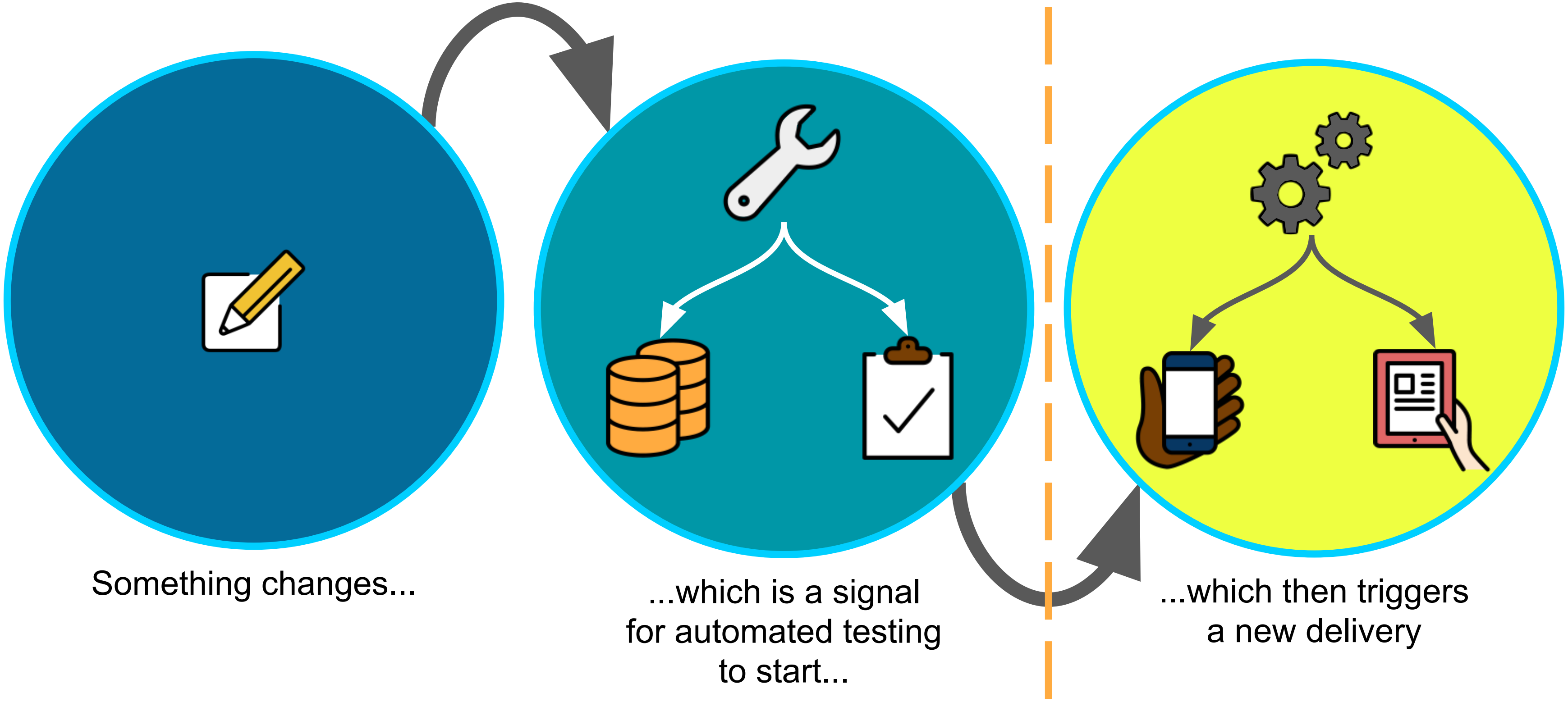


Something changes...

...which is a signal
for automated testing
to start...

...which then triggers
a new delivery

Continuous delivery

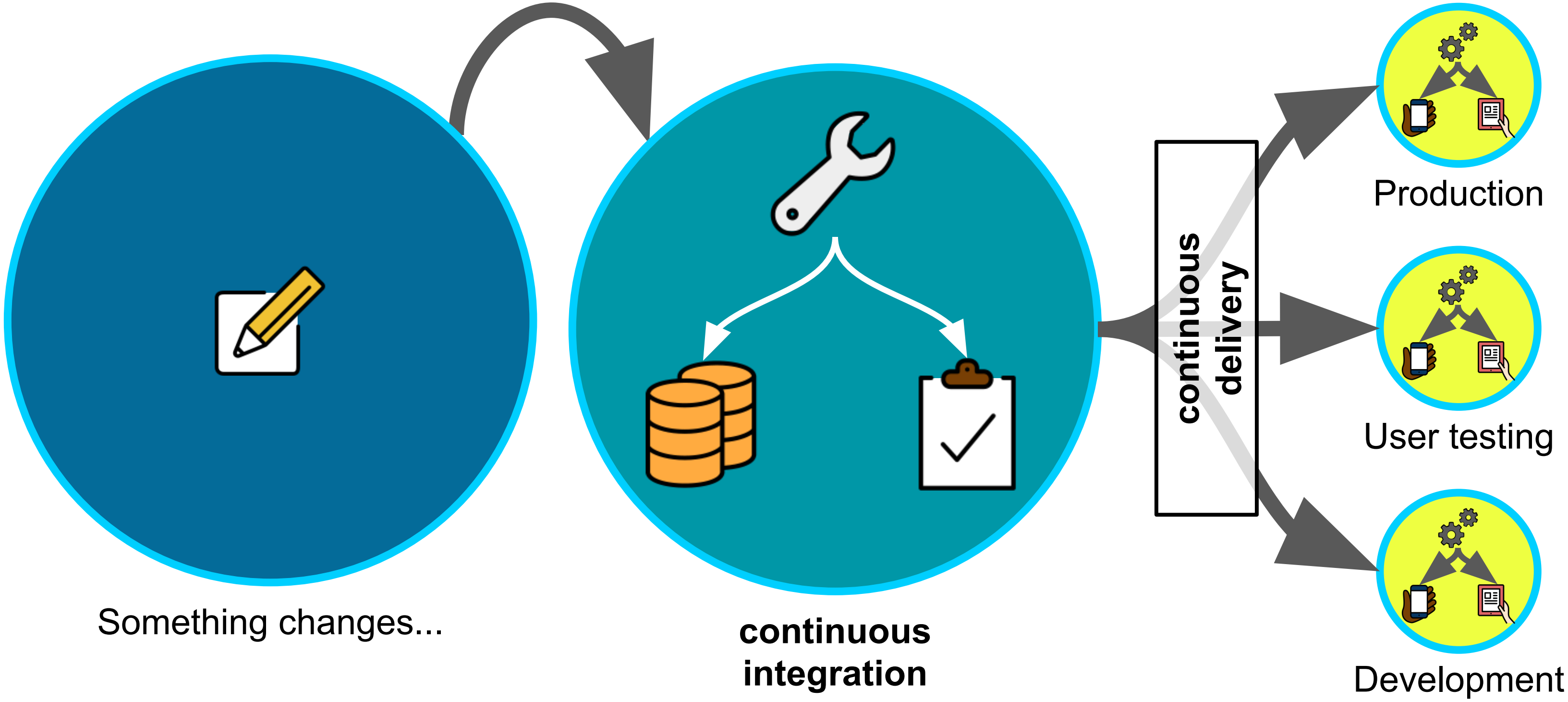


Something changes...

...which is a signal
for automated testing
to start...

...which then triggers
a new delivery

DevSecOps pipeline



Something changes...

continuous integration

continuous delivery

Production

User testing

Development

How does all this help?

Continuous delivery is making the latest changes available as often as possible

Continuous delivery is about frequently delivering small pieces of your product. This enables continuous user research and feedback to steer the product direction.

Continuous integration is testing everything, constantly, to catch problems early

Continuous integration should increase your confidence that the product is working as expected and not regressing unexpectedly. The regular, integrated security scans reduce your risk of a security incident and may reduce your compliance burden.

Automation is doing the same thing, the same way, every time

Automation reduces the risk of a mistake bringing whole systems down by ensuring that the delivery process is consistent.

How do we do it?

Someone on your team needs to be fluent in software development practices

All of the automation in the world is meaningless if you don’t understand the outputs. To review code, interpret tests, and provide feedback on technical direction, you need a tech lead. Without them, you won’t know when you’re off course or how to correct it.

What makes a good tech lead

The tech lead understands software, product management, user experience, agile, and user-centered design. They are not necessarily experts at any of it but they know enough to identify “good” and “bad.”

Build your DevSecOps pipeline at the start and insist that your teams use it

The sooner you have a pipeline, the sooner you can have small, frequent deployments, which makes it easier to make changes and helps the project move ahead smoothly. In healthy software development projects, the DevSecOps pipeline is often the very first thing that gets built.

Major components of a DevSecOps pipeline

- **Version control**, so you know when changes have happened
- **Continuous integration** platform, to execute your tests and let you review changes
- **Infrastructure defined in code**, so deliveries are repeatable and auditable
- **Testing and monitoring**, so you know if you know a change is going to work before you deliver it

Key takeaway:

DevOps enables **continuous, iterative development**

Software Development Practices

Greg Walker
18F