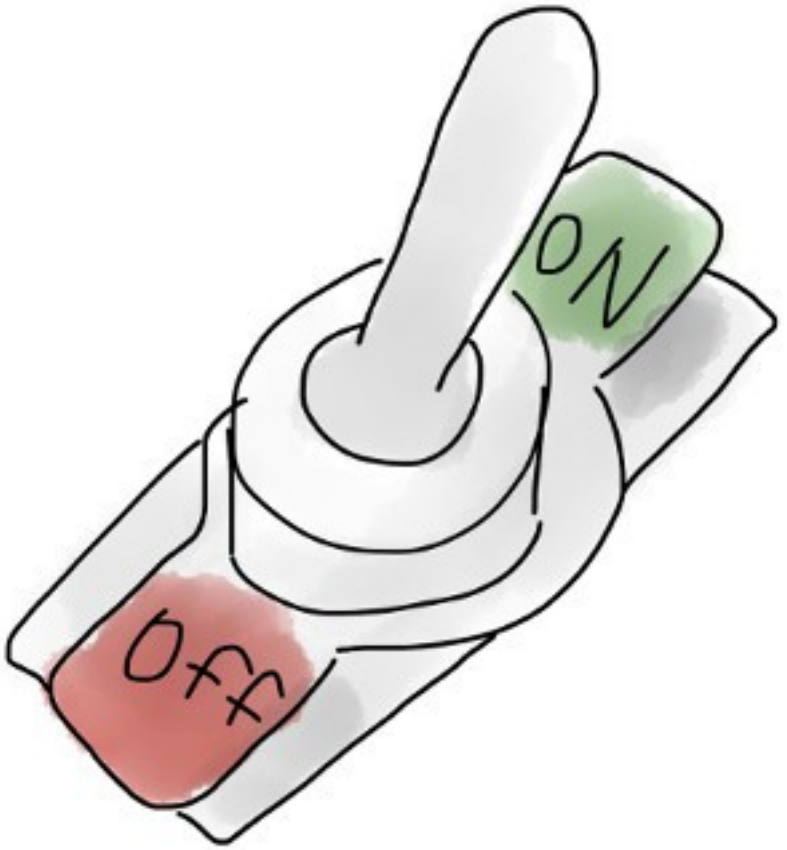
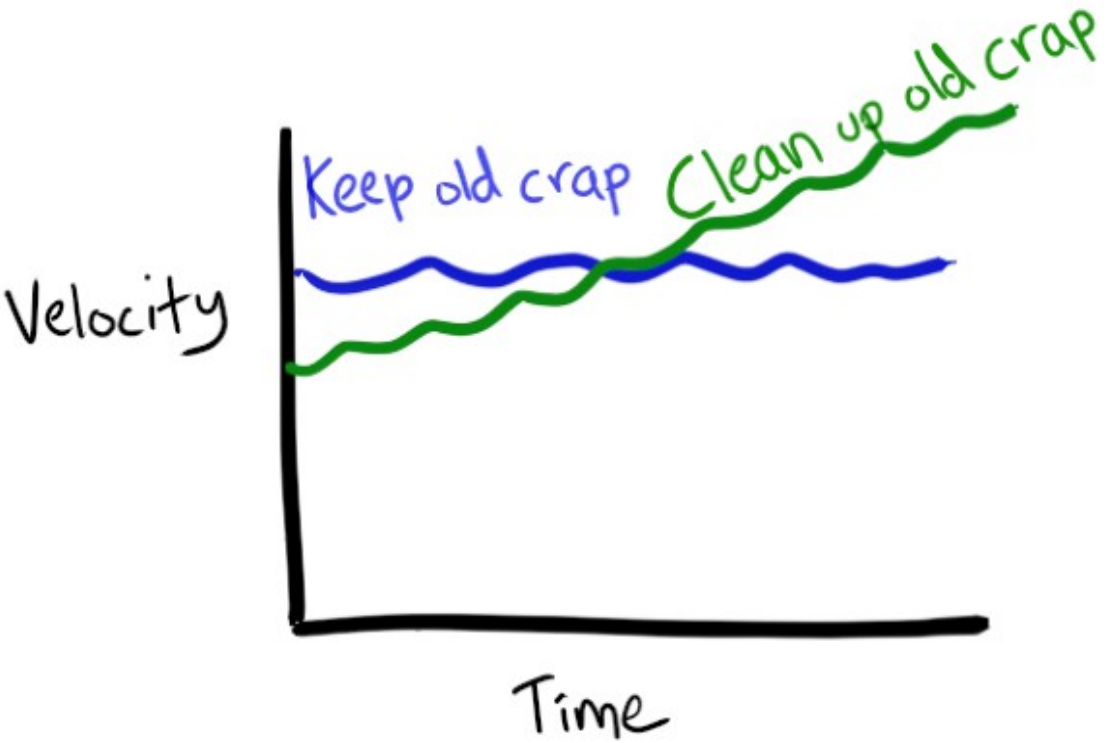
Feature Toggling: Control Over Code and Features During Release



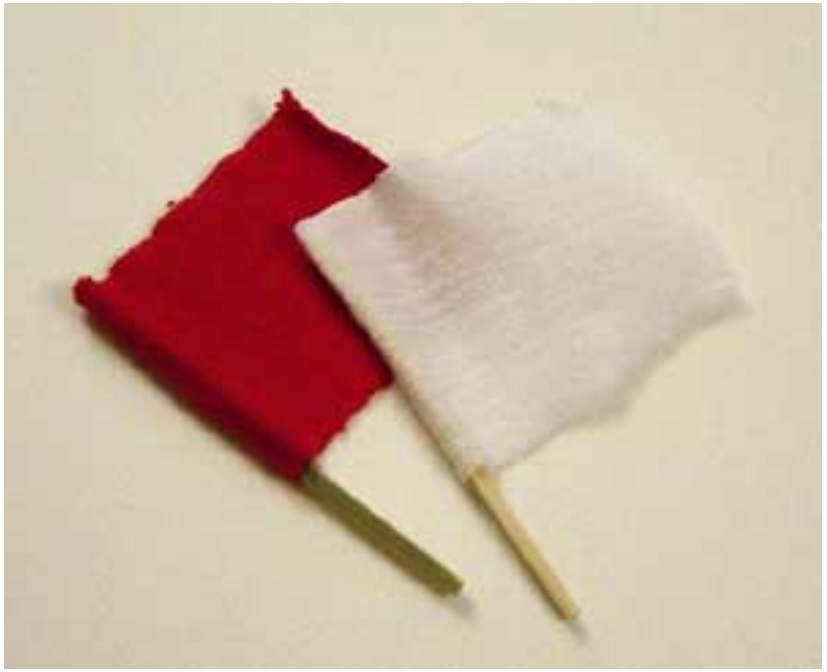
Md Tajmilur Rahman

Feature Flags are Technical Debts?

-Jim Bird


Keep old crap
Clean up old crap
Velocity
Time


fineart
america



TO STOP TRAFFIC

TO LET
TRAFFIC PROCEED

TO ALERT AND
SLOW TRAFFIC

Feature Flags/Switches/Bits

Basic Concept:

1. A configuration file

2. Use these flags in the code of the running application

3. Block the entry points to the features by toggling the corresponding flag

4. Unblock the feature just by toggling in the config file

** For feature toggles it's usually sufficient to run two combinations,

1. All the toggles on that are expected to be on in the next release

2. All toggles on

- Martin Fowler

Feature Flags/Switches/Bits

```
namespace switches {
    extern const char kDebugOnStart[];
    extern const char kEnableOpenMax[];
    extern const char kNoMessageBox[];
}
// Enable hardware decoding using OpenMax API.
// In practice this is for ChromeOS ARM.
const char kEnableOpenMax[] = "enable-openmax";
```

```
public void someBusinessMethod() {
  if( MyFeatures.FEATURE_ONE.isActive() ) {
      // do new exciting stuff here
  }
  [...]
}
```

Use Feature Flags with Caution

Feature Flags need to be Short Lived

Research Questions

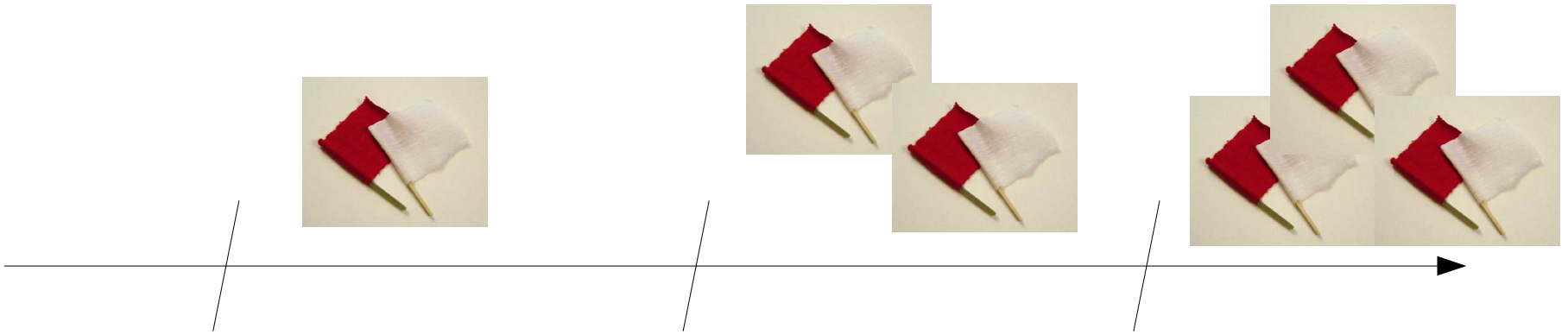
RQ1

What is feature flag, how feature flags are being used?

- Quantitative analysis to understand the usage of Flags and Toggles

RQ2

How many features get toggled in a release and how long lived are they?



RQ3

What percentage of code that is under feature toggles. How many are reverted right before a release / during stabilization?

```
1  Let S be the set of explored nodes with downlink route constructed
2  Initially S = g ∪ V_AP
3  Initially for each AP i in S, set Gi = ({g ∪ i}, {e_g,i}) and R_i = G_i
4
5  while S ≠ V do
6     Find S' ⊆ V-S: ∀ v ∈ S', v has at least two edges from S
7     // S_r is the reliable node set in S', initially S_r = ∅
8     if S' ≠ ∅ then
9           for all nodes v ∈ S' do
10                for all edge pairs (e_u1,v, e_u2,v) from S do
11                      h_u1,u2 = (h_u1 + h_u2)/2
12                end for
13                Find Pv, set of edge pairs of v satisfying C1 ∧ (C2 ∪ C3)
14                if Pv ≠ ∅ then
15                      Sr = Sr ∪ {v}
16                      Choose (e_u1,v, e_u2,v) from P_v with min h_u1,u2
17                else
18                      Choose (e_u1,v, e_u2,v) from S' with min h_u1,u2
19                end if
20                h_v = h_u1,u2 + 1
21          end for
22          if Sr ≠ ∅ then
23                Add v in S_r with min h_v to S
24          else
25                Add v in S with min h_v to S
26          end if
27          ConstructDG(G, u_1, u_2, v);
28    else
29          Find S'' ⊆ V-S and ∀v ∈ S'', v has one edge e_u,v from S
30          if S'' ≠ ∅ then
31                for all node v ∈ S'' do
32                      h_v = h_u + 1
33                end for
34                Add v to S with min h_v
35                G_v = ({u ∪ v},{e_u,v})
36                R_v = R_u → G_v
37          else
38                return FAIL;
39          end if
40    end if
41 end while
42 return SUCCESS
```

CODE BLOCK 1

RQ4

Who are the developer(s) who do the toggling?

RQ5

What aspects lead developers to make feature toggles?