

@bctechdays.com

Thu 22 Jun, 11:30 - 13:00

Feature Management for Continuous Delivery



Tharanga
Chandrasekara



Vlad
Leonov



**BC/NAV
TECH
DAYS
2023**

mibuso.com

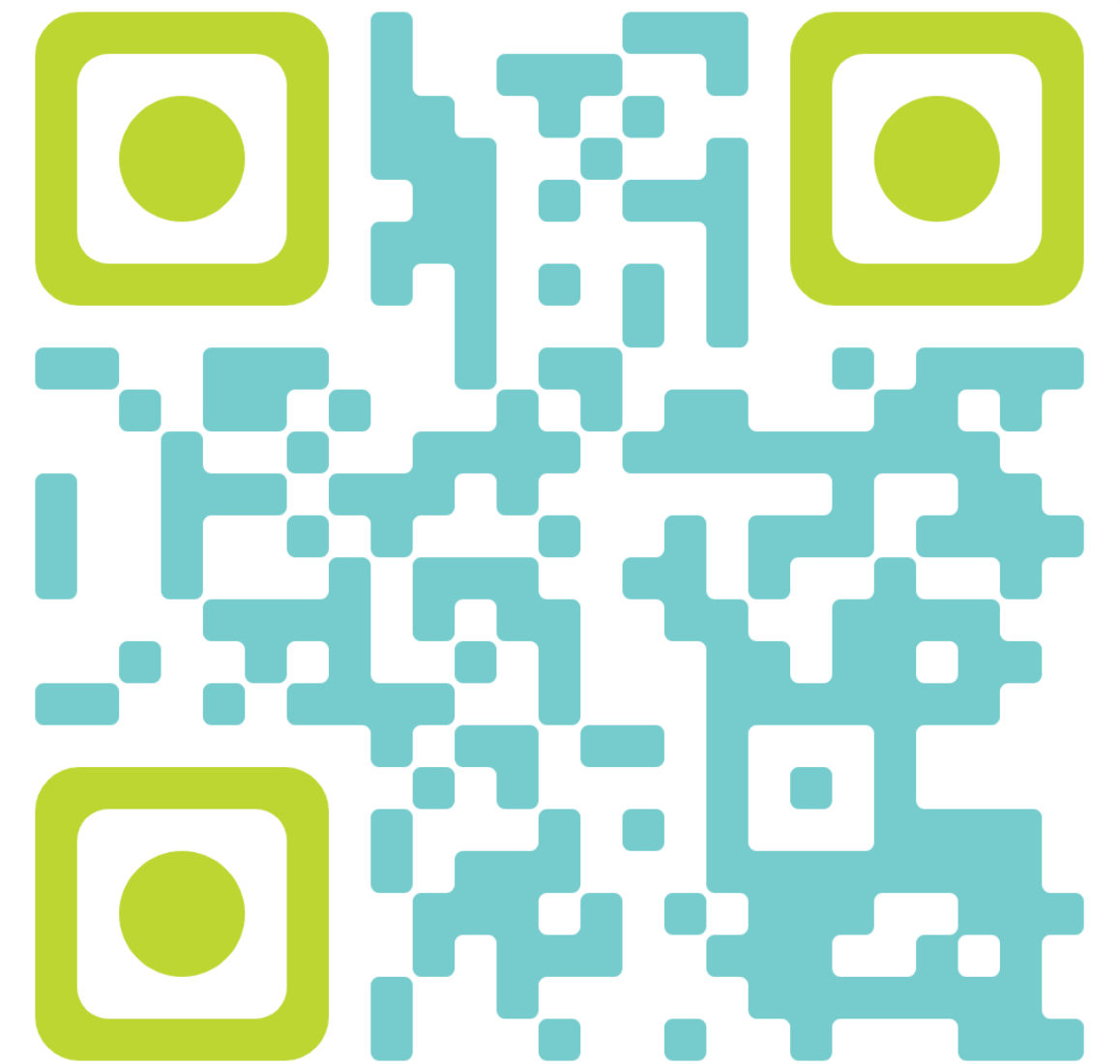


Tharanga Chandrasekara

Lead Technical Consultant at  theta

- Microsoft MVP for the past seven years
- Lives in Auckland, New Zealand 
-  Made in Anuradhapura, Sri Lanka
- twitter.com/TharangaNC



github.com/TharangaC



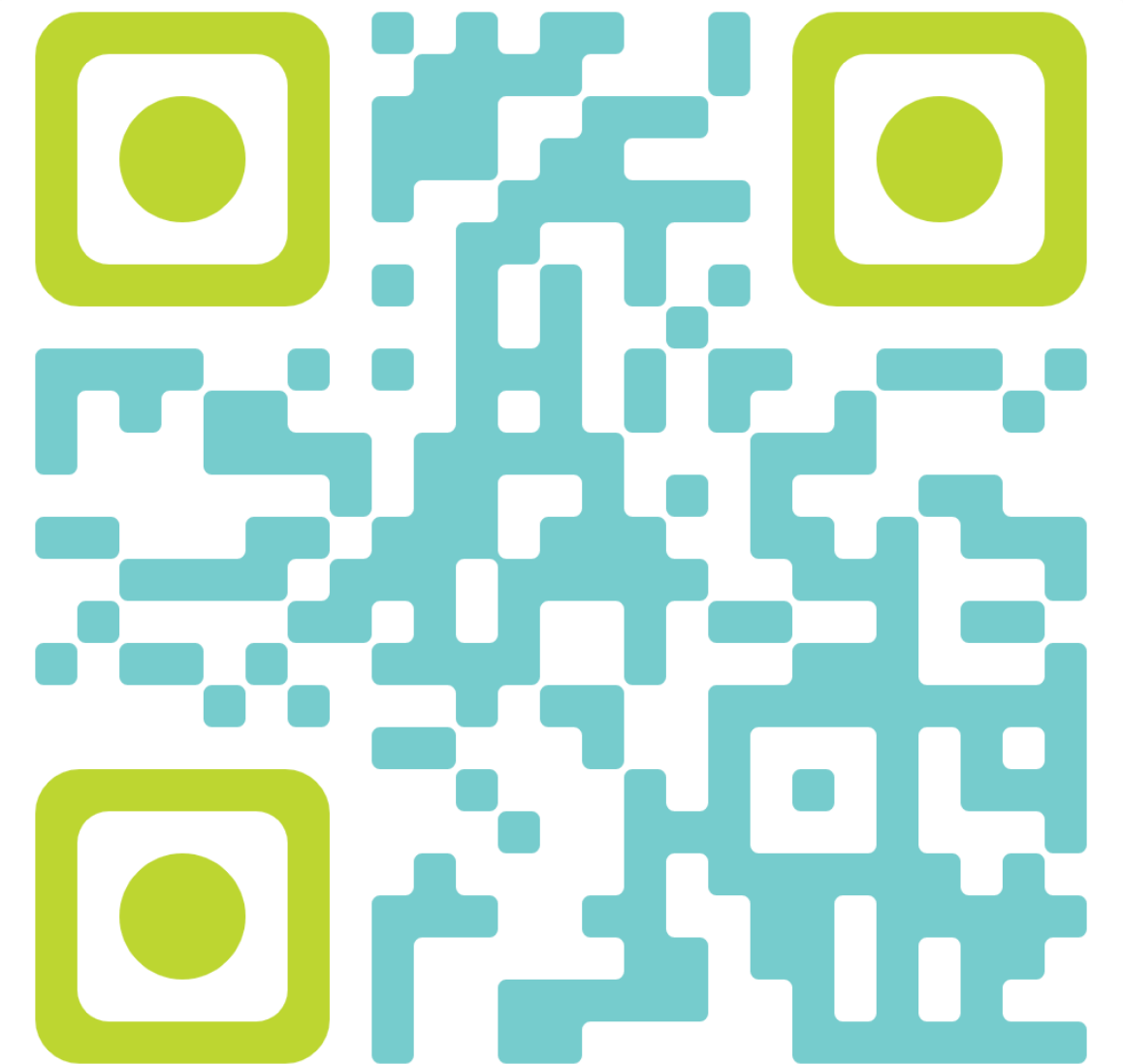


Vlad Leonov

Technical Lead at  theta

- Lives in Christchurch, New Zealand 
-  Made in Kyiv, Ukraine
- twitter.com/vodyl

github.com/vody



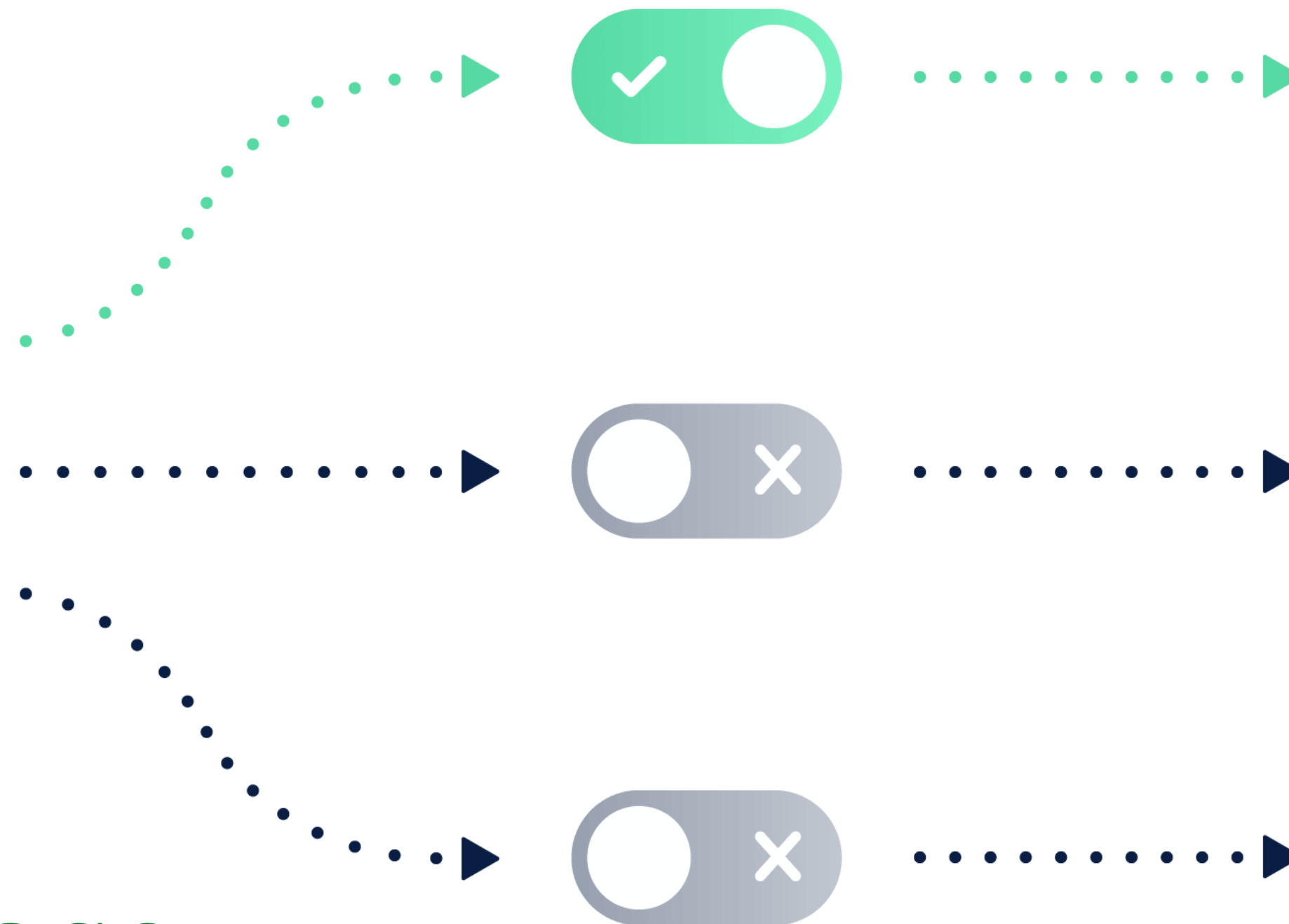
What is a Feature Flags?

aka. feature switchers, feature toggles, release toggles or feature flippers

New feature

Feature Flags

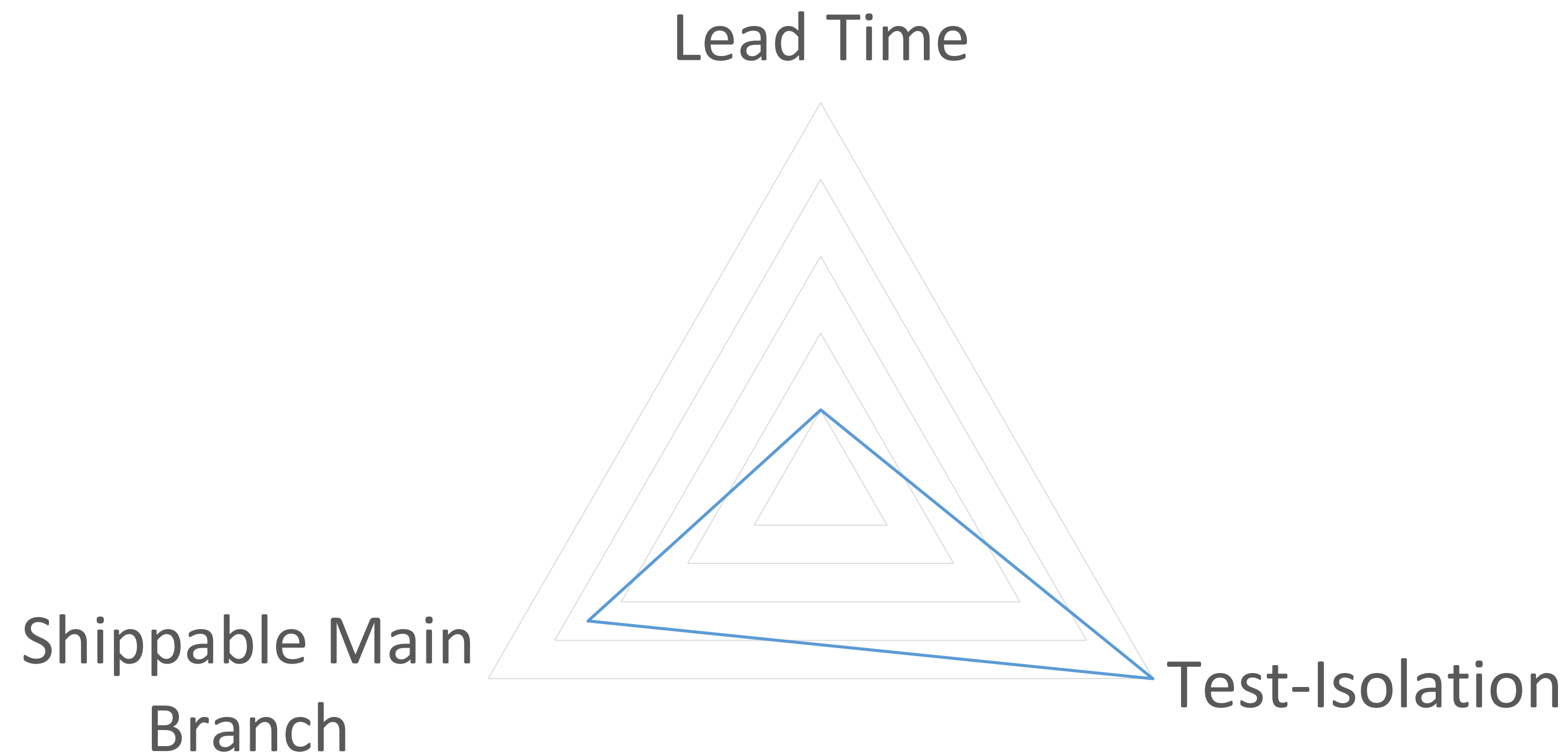
Customers



Deploy \neq Release

Use-Cases

- Continues Integration



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	<div>A 2</div> <div>B 2</div>				
BLOCKED					

The team is working on Feature A and B in parallel within separate feature branches.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	<div>B 2</div>	<div>A 2</div>			
BLOCKED					

Feature A coded first and integrated into main branch



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	B 2		A 2		
BLOCKED					

Feature A moved to a test environment, where functionals can validate it in isolation.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div>A 2</div>		
BLOCKED		<div>B 2</div>			

Feature B is ready for integration but **blocked** until Feature A test is complete.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div>A 3</div>		
BLOCKED		<div>B 2+1</div>			

Feature B is ready for integration but **blocked** until Feature A test is complete.



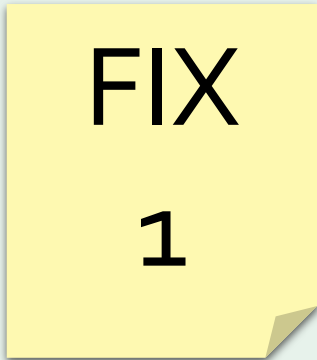
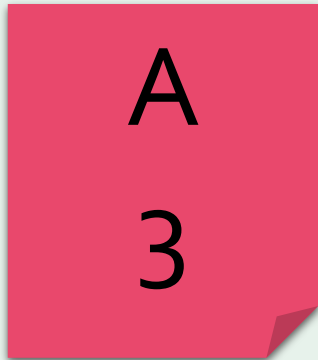
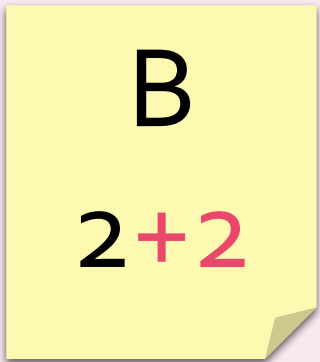
Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div>A 3</div>		
BLOCKED		<div>B 2+1</div>			

Feature A **test failed**. The team decided to **wait** for a quick fix instead of a rollback.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE					
BLOCKED					

Feature A fix is on its way. Feature B is **still blocked**.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE		<div>FIX 1</div>	<div>A 3</div>		
BLOCKED		<div>B 2+2</div>			

Feature A fix is on its way. Feature B is **still blocked**.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div data-bbox="1649 709 1855 943"> A 4 </div>		
BLOCKED		<div data-bbox="1122 1123 1329 1358"> B 2+2 </div>			

Feature A **fix finally arrived** at the test environment.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div>A 5</div>		
BLOCKED		<div>B 2+3</div>			

Feature A on re-test. Feature B is **still blocked**.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE				<div>A 5</div>	
BLOCKED		<div>B 2+3</div>			

Feature A **test passed** and moved to UAT. Feature B **can be integrated**.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE		<div>B 2+3</div>		<div>A 5</div>	
BLOCKED					

Feature A **test passed** and moved to UAT. Feature B **can be integrated**.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div>B 2+3</div>	<div>A 5</div>	
BLOCKED					

Feature B is ready for testing.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div>B 3+3</div>	<div>A 6</div>	
BLOCKED					

Feature B **test passed**. But Feature A **was not accepted** on UAT and required a rework!



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE				<div>A 6</div>	
BLOCKED			<div>B 3+3</div>		

Feature B now **needs to wait** for Feature A rollback.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	<div>A 6</div>				
BLOCKED			<div>B 3+3</div>		

Feature B now **needs to wait** for Feature A rollback.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	<div>A 7</div>				
BLOCKED			<div>B 3+4</div>		

Feature B now **needs to wait** for Feature A rollback.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	<div>A 7</div>		<div>B 3+4</div>		
BLOCKED					

Feature B arrived on the test.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	<div>A 7</div>		<div>B 4+4</div>		
BLOCKED					

Feature B test passed.



Delivery Pipeline **without** feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	<div>A 7</div>			<div>B 4+4</div>	
BLOCKED					

Feature B finally got to UAT for user acceptance testing.



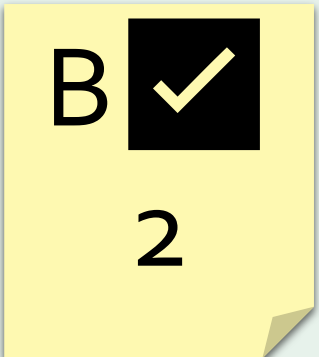
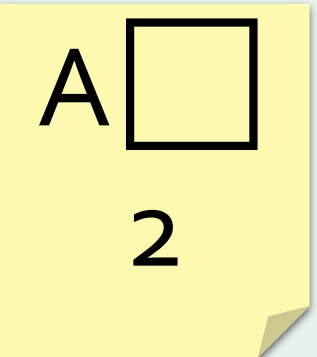
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	<div> <div>A <input checked="" type="checkbox"/> 2</div> <div>B <input checked="" type="checkbox"/> 2</div> </div>				
BLOCKED					

The team is working on Feature A and B in parallel within separate feature branches.



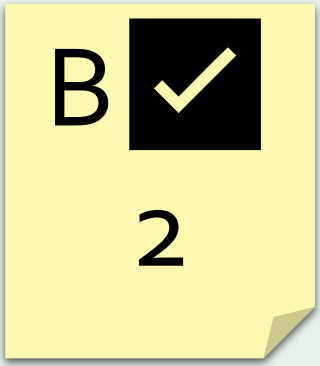
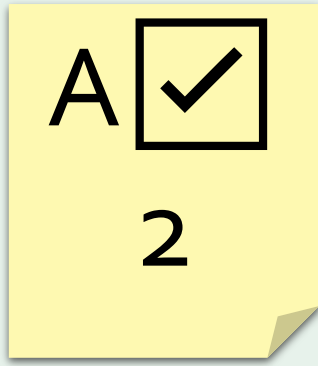
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE					
BLOCKED					

Feature A coded first and integrated into main branch



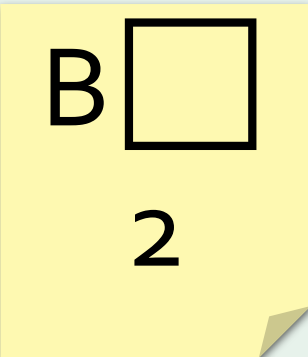
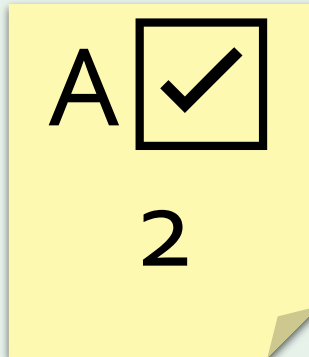
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE	 B <input checked="" type="checkbox"/> 2		 A <input checked="" type="checkbox"/> 2		
BLOCKED					

Feature A moved to a test environment, where functionals can validate it in isolation.



Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE		 2	 2		
BLOCKED					

Feature B code is done. Because Feature B is behind a Feature Flag, we can safely integrate it without being wary of interrupting ongoing testing.



Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div>A <input checked="" type="checkbox"/> 2</div> <div>B <input checked="" type="checkbox"/> 2</div>		
BLOCKED					

Feature A and B can be tested in isolation within the same environment in parallel without redeployment.



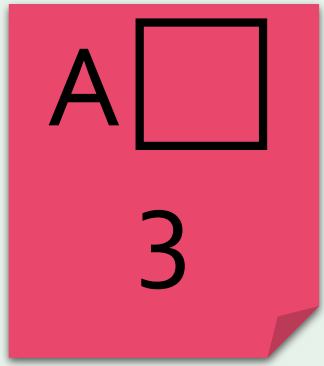
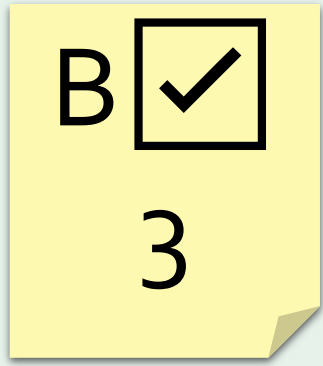
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			<div>A <input checked="" type="checkbox"/> 3</div> <div>B <input checked="" type="checkbox"/> 3</div>		
BLOCKED					

Feature A and B can be tested in isolation within the same environment in parallel without redeployment.



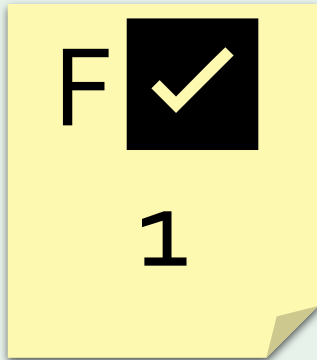
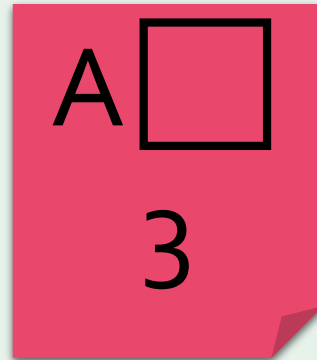
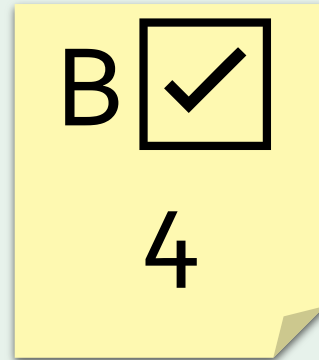
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE					
BLOCKED					

Feature A **test failed**. Feature B **test passed** and can be pushed to UAT.



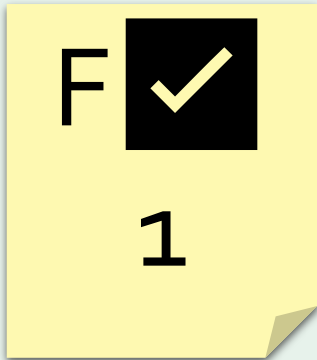
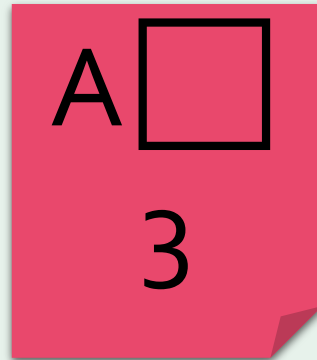
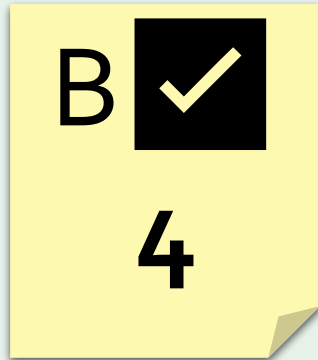
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE					
BLOCKED					

Because broken Feature B is behind a feature flag, **we can proceed** with Feature B UAT.



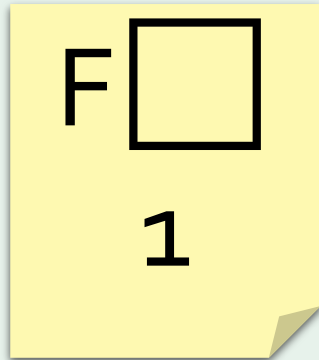
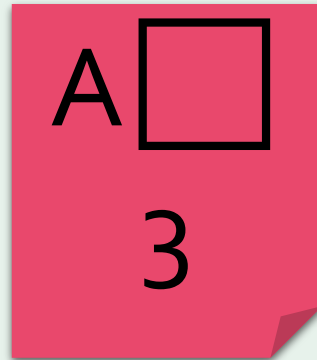
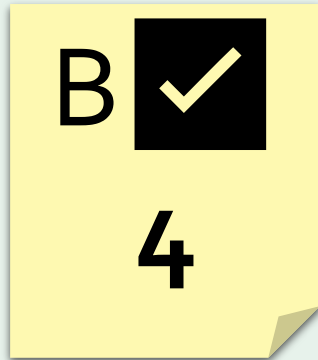
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE					
BLOCKED					

Feature B UAT acceptance testing passed and moved to LIVE! We can keep Feature A included in our release... because... Yes, it's disabled by a feature flag.



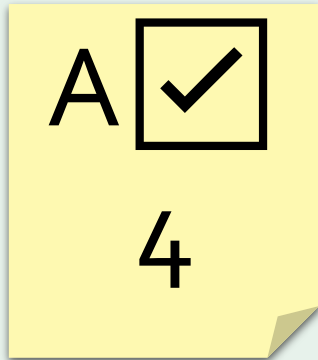
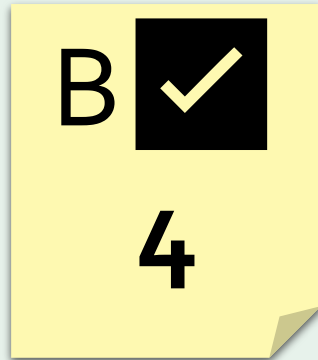
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE					
BLOCKED					

Feature B is **already live**, while Feature A fix is coming.



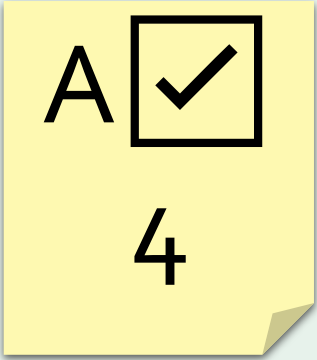
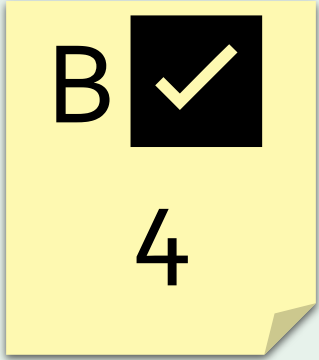
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE			 A <input checked="" type="checkbox"/> 4		 B <input checked="" type="checkbox"/> 4
BLOCKED					

Feature B is **already live**, while Feature A fix is coming.



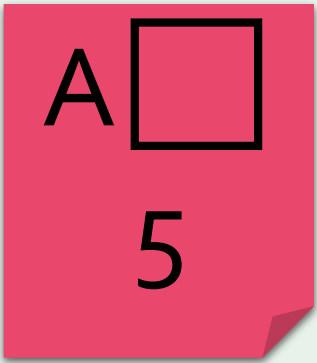
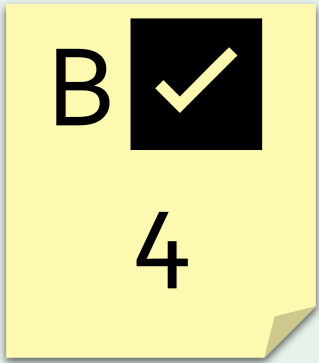
Delivery Pipeline with feature flags

	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE					
BLOCKED					

Feature A arrived at UAT



Delivery Pipeline with feature flags

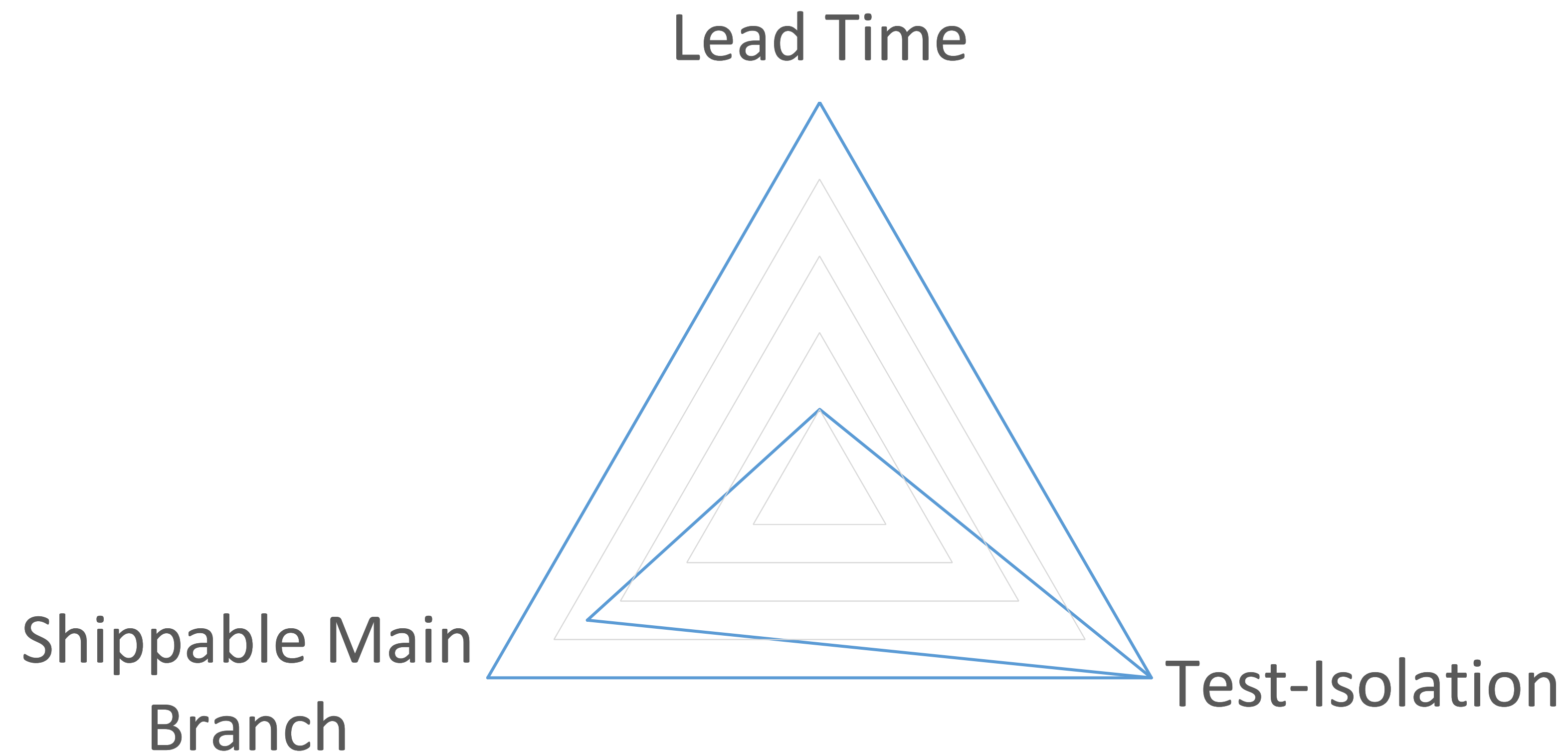
	CODE	INTEGRATE	TEST	UAT	LIVE
ACTIVE					
BLOCKED					

Feature A **was not accepted** on UAT and required a rework



Use-Cases

- Continues Integration



Use-Cases

- Continues Integration
- Control Rollout, aka “Progressive Delivery”
- Kill Switch
- Running Experiments, aka “A/B Testing”
- User Targeting
- Launch on Time
- Early Access

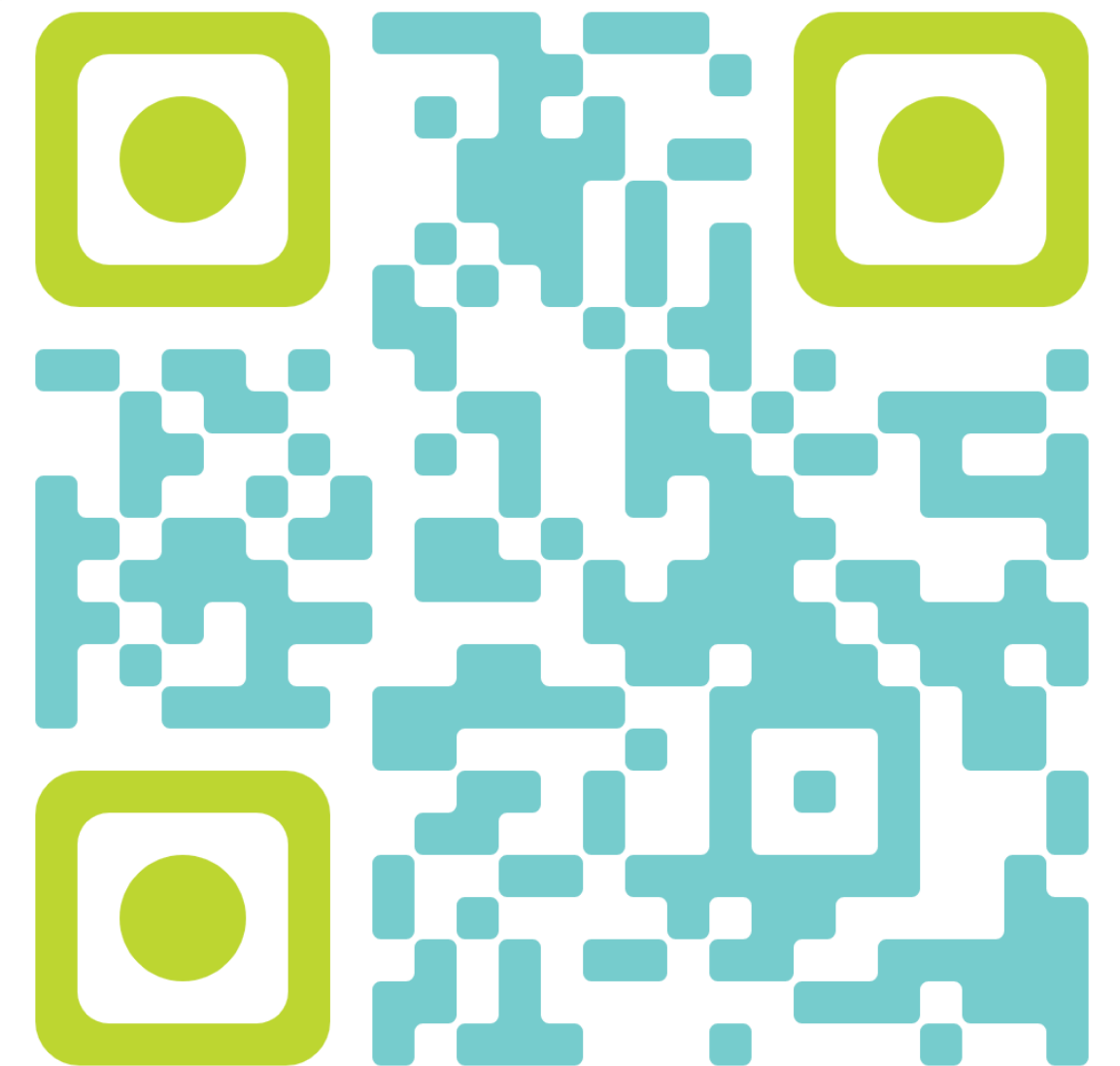


Microsoft Feature Management

Enabling Upcoming Features Ahead of Time

- Limit the dramatic changes to user experience.
- Improve the stability of the SaaS platform.
- Decouple user training with upgrades.

365go.in/FeatureMgmt

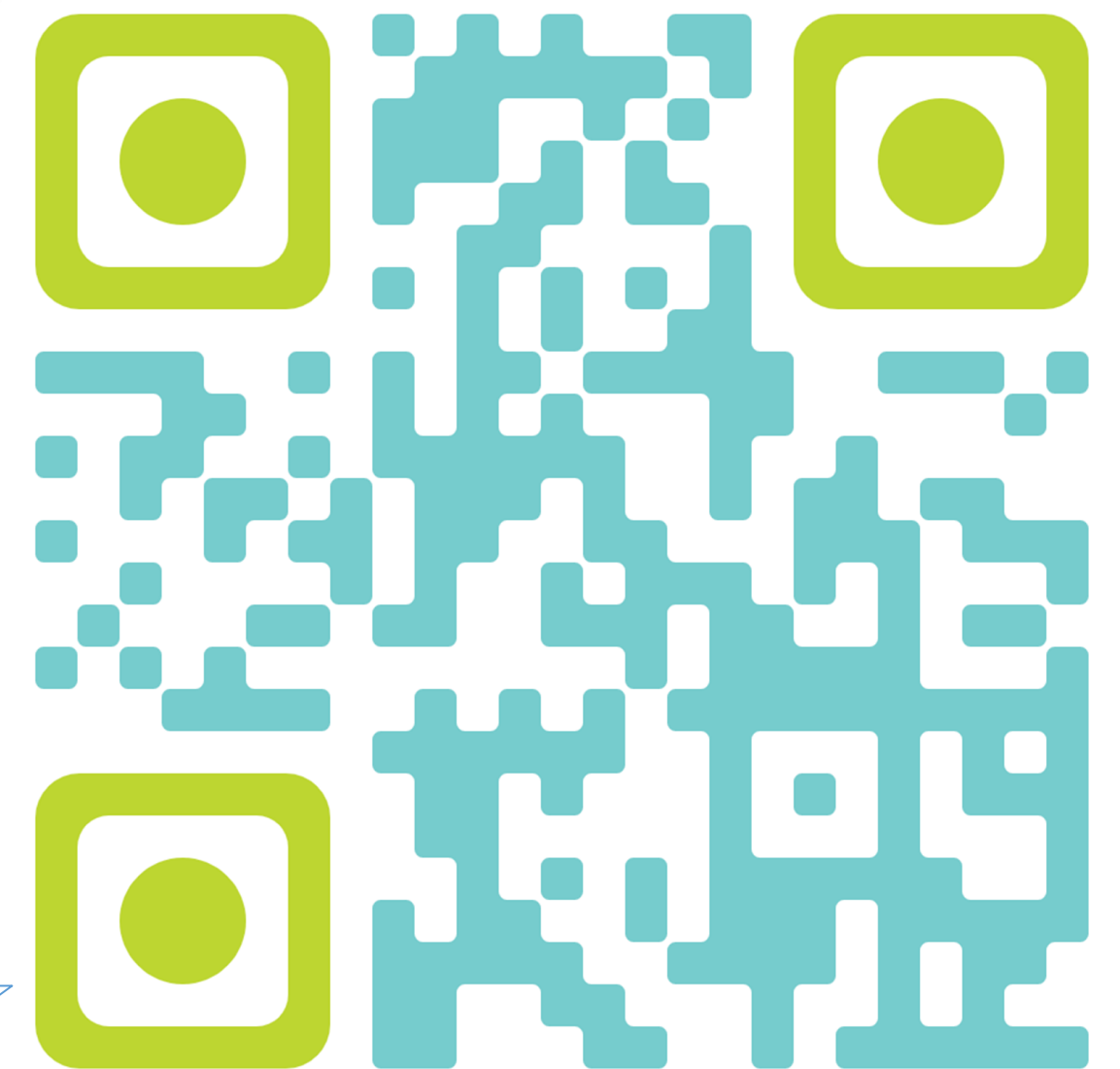


Microsoft Feature Management

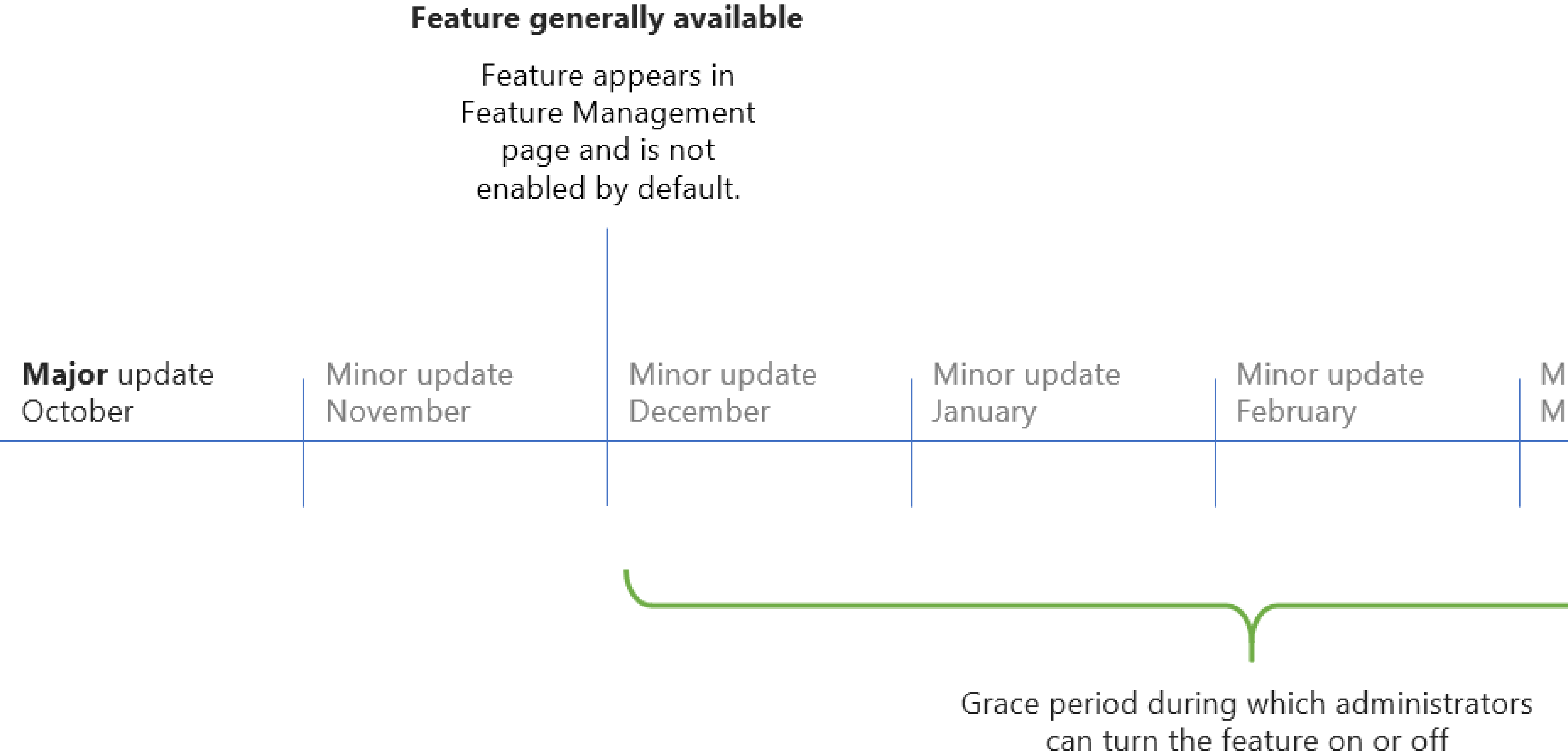
How to be on top of the latest features from Microsoft?

- Insider Build
- Release plan
- Roadmap

365go.in/BCFeatures



Microsoft Release Cycle



Telemetry

Analyze and Monitor Telemetry with Power BI

Article • 04/14/2023 • 4 contributors

In this article

- About the reports in the app
- Get the apps
- Connect to Azure Application Insights for the first time
- Configure an app after initial setup
- Show 5 more

To make it simple to analyze Business Central AppSource. One app is for telemetry on environment (defined in app.json). Both apps are free and available on the AppSource.

About the reports

The app consists of four reports:

- Usage
- Errors
- Performance
- Administration



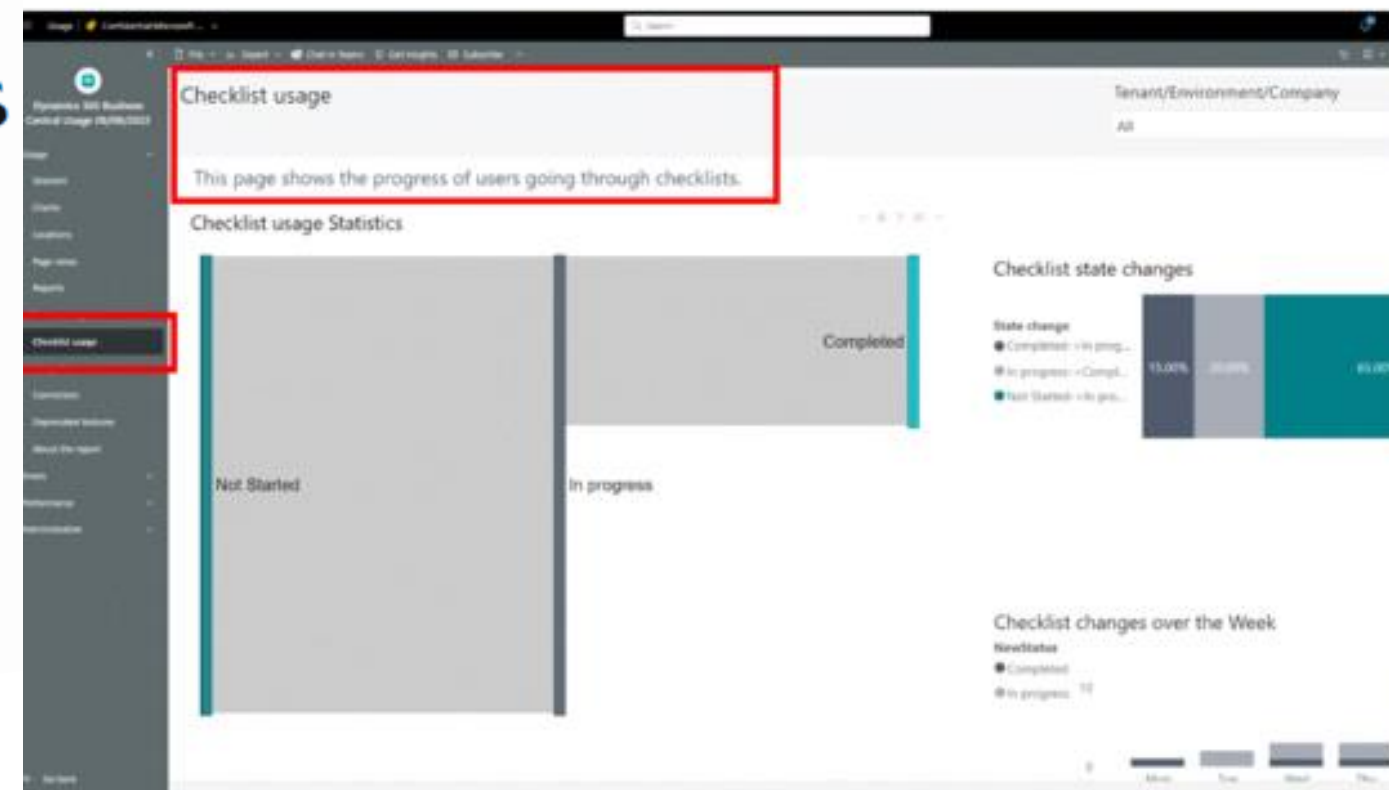
Kennie Nybo Pontoppidan • 1st
Principal Program Manager at Microsoft
9mo •

New article about Dynamics 365 Business Central telemetry. This is also the first article in a new newsletter on this topic.

...see more



Business Central Telemetry



What's new in Dynamics 365 Business Central telemetry - October 2022

Kennie Nybo Pontoppidan on LinkedIn • 8 min read



Kennie Nybo Pontoppidan • 1st
Principal Program Manager at Microsoft
7mo •

...

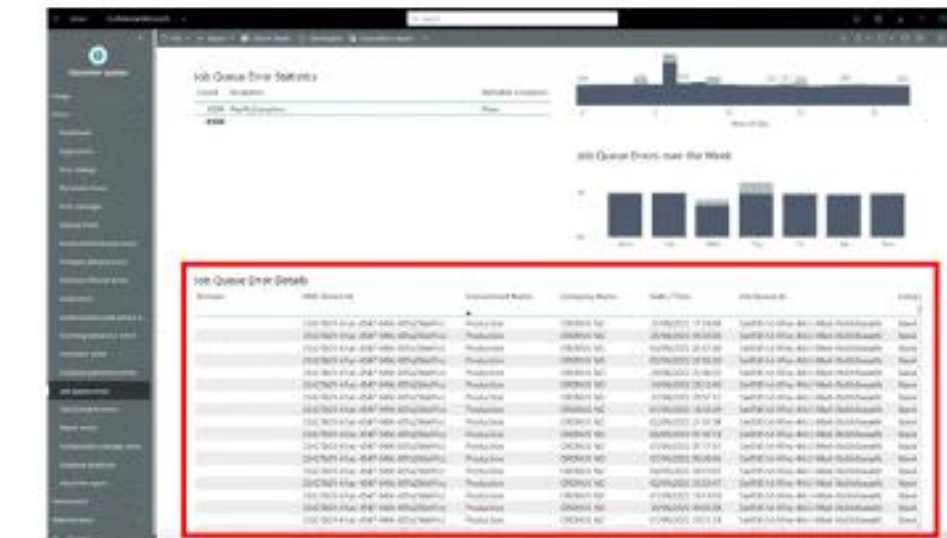
It is now December 1st in New Zealand, so the December newsletter on Business Central telemetry is now live. Hope you learn a thing or two.



Business Central Telemetry

+ Subscribe

Error Report: Job Queue Errors page
– get all details of errors (and maybe export to Excel?)



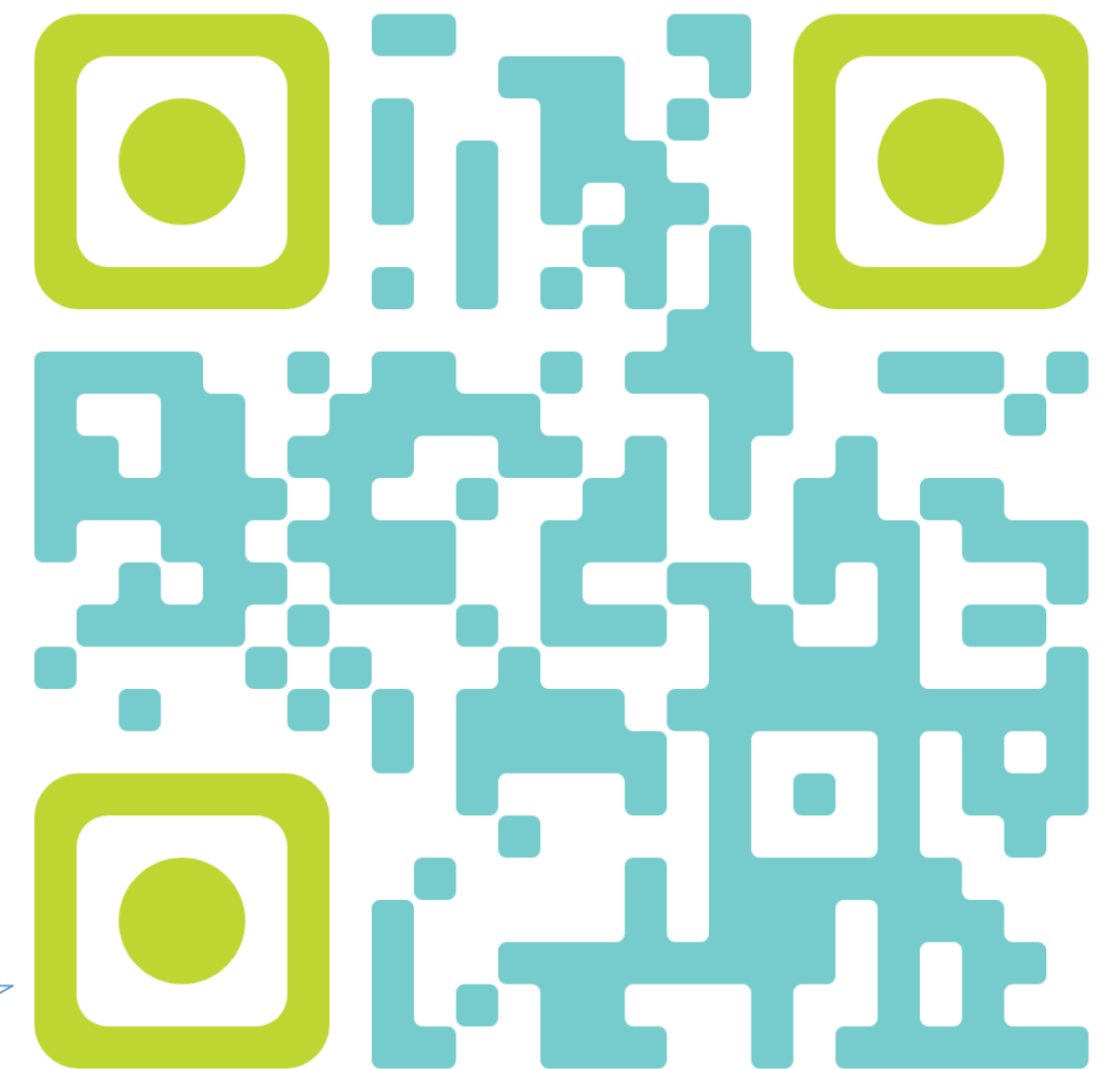
What's new in Dynamics 365 Business Central telemetry - December 2022

Kennie Nybo Pontoppidan on LinkedIn • 12 min read

Feature Aware Apps

- Feature-aware apps leverage feature flags to adapt their functionality and user experience dynamically.
- Provides partners with flexibility, control, and the ability to deliver features to users in a controlled manner.

365go.in/FeatureAwareApp



How to Check Feature is Enabled

```
1 procedure SetUserPermissionProvider()  
2 var  
3     FeatureManagementFacade: Codeunit "Feature Management Facade";  
4     FeatureFlaglbl: Label 'HideLegacyUserGroups';  
5 begin  
6     if FeatureManagementFacade.IsEnabled(FeatureFlaglbl) then  
7         permissionProviderFeature := permissionProviderFeature::Modern  
8     else  
9         permissionProviderFeature := permissionProviderFeature::Legacy;  
10 end;
```



Feature Management Implementation

```
1 codeunit 2611 "Feature Management Facade"  
2 {  
3     Access = Public;  
4  
5     var  
6         FeatureManagementImpl: Codeunit "Feature Management Impl.";
```

```
1 procedure IsEnabled(FeatureId: Text[50]): Boolean;  
2 var  
3     FeatureKey: Record "Feature Key";  
4     FeatureDataUpdateStatus: Record "Feature Data Update Status";  
5 begin  
6     if FeatureKey.Get(FeatureId) then begin  
7         InitializeFeatureDataUpdateStatus(FeatureKey, FeatureDataUpdateStatus);  
8         exit(FeatureDataUpdateStatus."Feature Status" in ["Feature Status"::Complete, "Feature Status"::Enabled])  
9     end;  
10 end;
```



How to implement Feature Flags?

Process

- Choose the Right Level of Flagging:
Epic, Feature, Story
- Isolation Check: validate each commit to confirm feature isolation
- Iterations: limit feature lifecycle
- Feedback Loop: boost customer engagement
- Culture of Experimentation:
MINDSET_CHANGE_WARNING

Tools

- Microsoft Feature Management: only for Platform Team?
- DIY: <MyFeature> Boolean in Setup
- OpenFeature for AL: 365go.in/FF



OpenFeature for AL

release v3.2.0.0

last commit yesterday

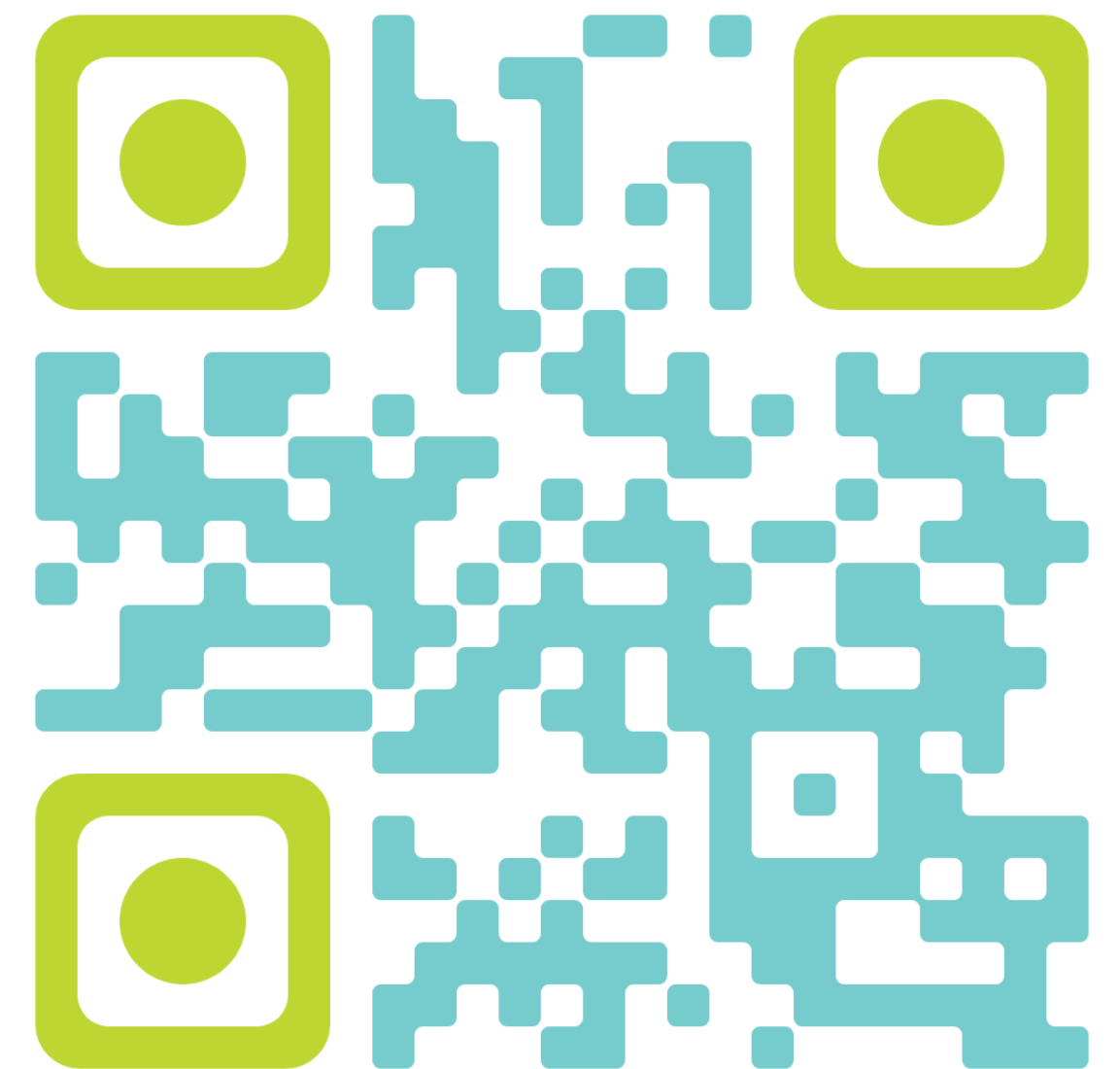
open issues 2

open pull requests 1

license GPL-3.0

 Tweet

```
1 field(RewardLevel; RewardLevel)
2 {
3     ApplicationArea = CustomerRewards;
4     Caption = 'Reward Level';
5     Editable = false;
6 }
```

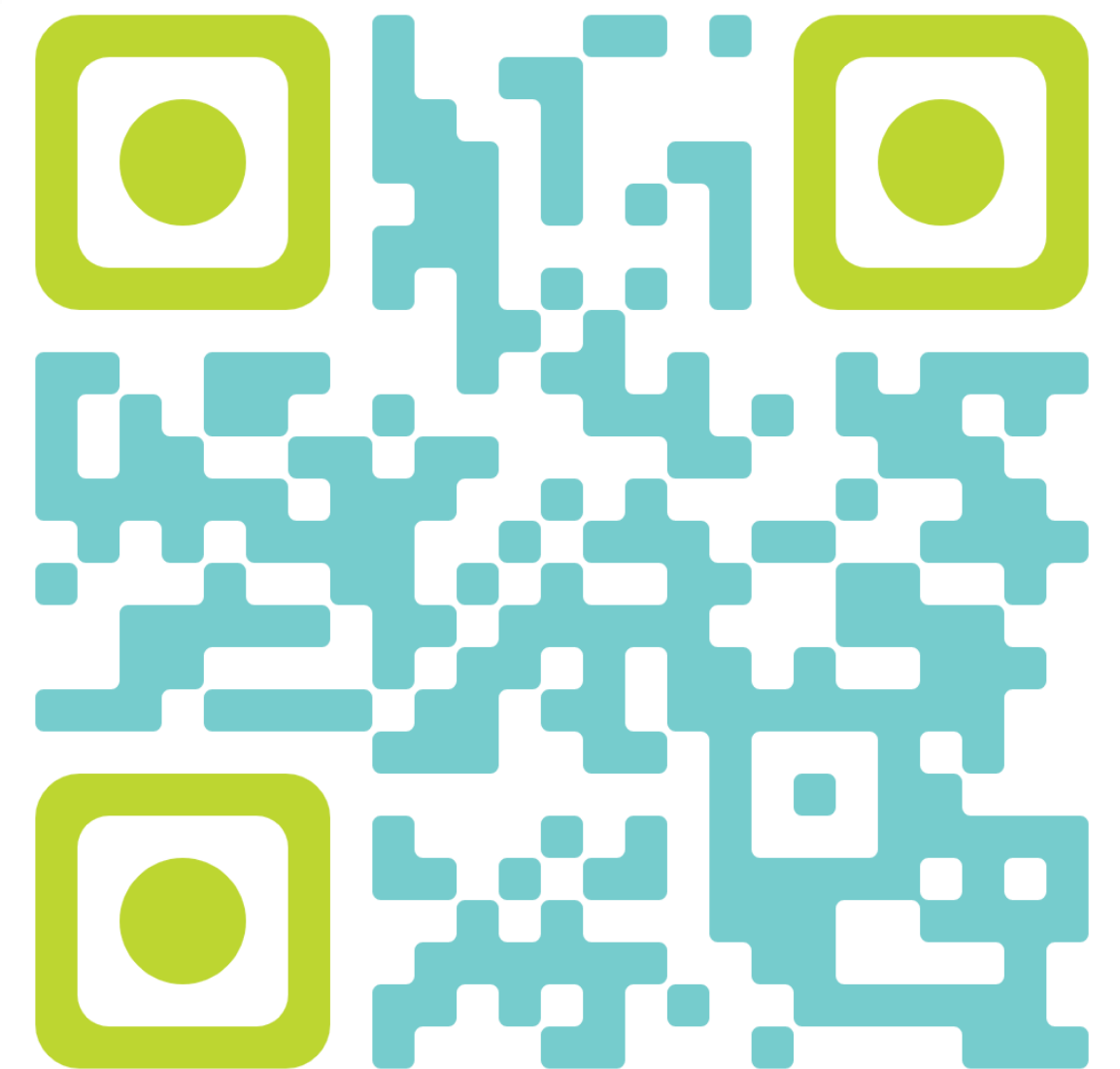


365go.in/FF

OpenFeature for AL Pages

Use your feature identifier as an *ApplicationArea* for all feature-related page controls and actions.

```
1 field(RewardLevel; RewardLevel)
2 {
3     ApplicationArea = CustomerRewards;
4     Caption = 'Reward Level';
5     Editable = false;
6 }
```



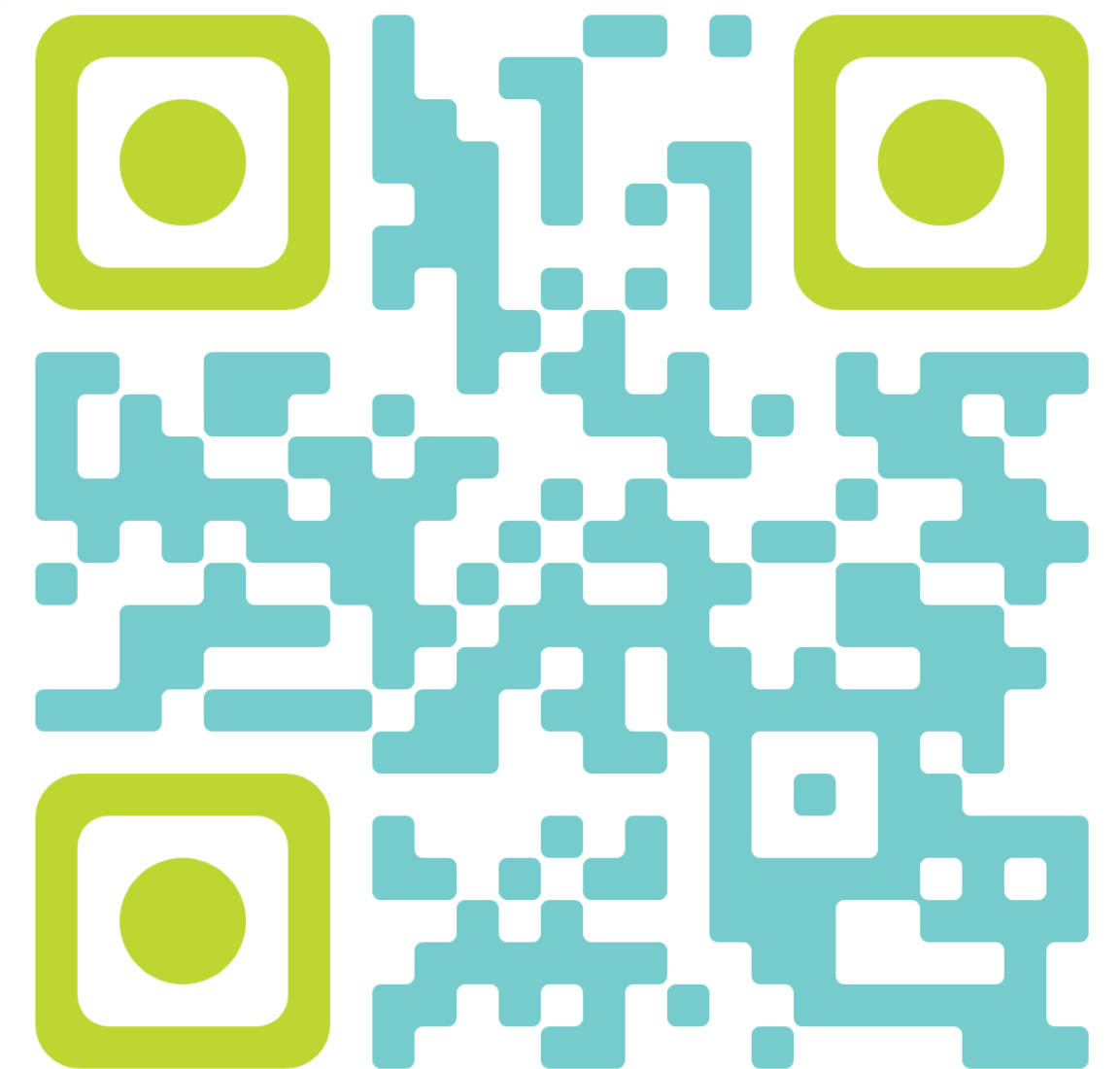
365go.in/FF



OpenFeature for AL Code

Wrap feature-related code block into a condition. It's recommended to use IsEnabled function from FeatureMgt_FF_TSL codeunit.

```
1 trigger OnAfterGetRecord();
2 var
3     CustomerRewardsMgtExt: Codeunit "Customer Rewards Ext. Mgt.";
4 begin
5     if StrPos(ApplicationArea(), '#CustomerRewards,') <> 0 then
6         RewardLevel := CustomerRewardsMgtExt.GetRewardLevel(RewardPoints);
7 end;
```



365go.in/FF



OpenFeature Providers

Conditional Provider

- Good For Per-Tenant Extension
- Feature defined by the extension itself.

```
1 trigger OnInstallAppPerDatabase()  
2 var  
3     ConditionProvider: Codeunit ConditionProvider_FF_TSL;  
4 begin  
5     ConditionProvider.AddFeature('CustomerRewards', '[More Info](https://example.com)');  
6 end;
```



OpenFeature Providers

Conditional Provider

- Good For Per-Tenant Extension
- Feature defined by the extension itself.
- Feature state conditions defined by the user in the customer environment.

Conditions

Search

+ New

Edit List

Delete

↶

⌵

☰

Code ↑	Function	Argument	Active
→ <u>EVALUATIONCOMP</u>	Company Filter	Evaluation Company=FILTER(true)	<input checked="" type="checkbox"/>
EVERYONE	User Filter		<input checked="" type="checkbox"/>

```

, Company.GetView());

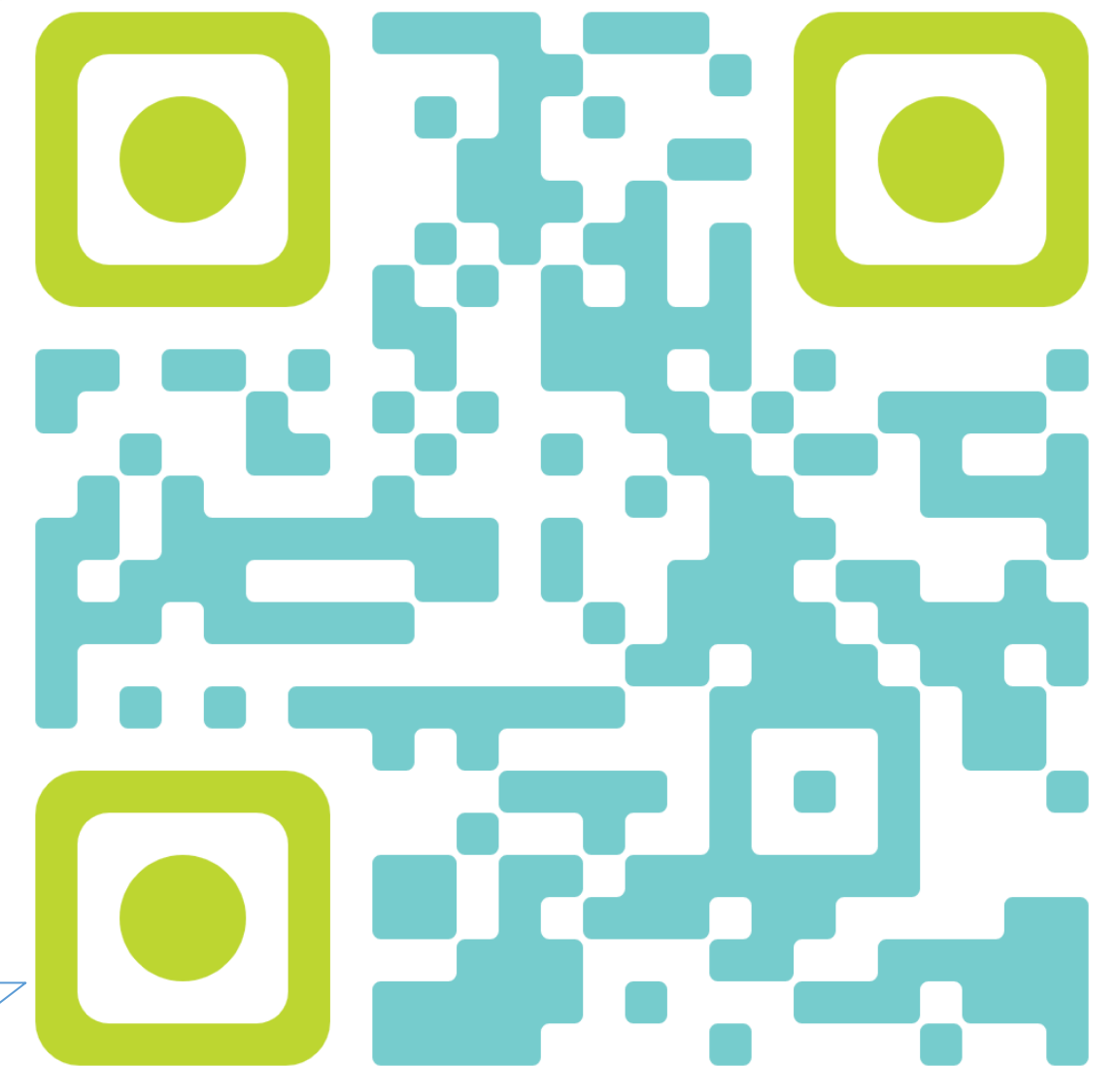
```

OpenFeature Providers

Conditional Provider

- Good For Per-Tenant Extension
- Feature defined by the extension itself
- Feature state conditions defined by the user in the customer environment




365go.in/FFExample



OpenFeature Providers

PostHog Provider

- Good For AppSource Extension



Feature flags

PostHog does that.

Safely roll out features to select users or cohorts.

[Get started - free](#)[View pricing](#)

KEY	CREATED BY	CREATED	RELEASE CONDITIONS	STATUS
<input type="checkbox"/> new-sessions-player Moves to the new playback experience for session recordings.	Burt	21 days ago	100% of all users	Enabled
<input checked="" type="checkbox"/> new-sessions-player Moves to the new playback experience for session recordings.	Burt	21 days ago	Multiple groups	Disabled
<input checked="" type="checkbox"/> new-sessions-player Moves to the new playback experience for session recordings.	Burt	21 days ago	100% of mibusingerations	

[View usage](#)[Copy feature flag key](#)[Edit](#)[Delete feature flag](#)



OpenFeature Providers

PostHog Provider

- Good For AppSource Extension
- Feature defined in PostHog and then propagated to all customers.

```
1 trigger OnInstallAppPerCompany()
2 var
3     PostHogProvider: Codeunit PostHogProvider_FF_TSL;
4     PersonalAPIKey, ProjectID : Text;
5 begin
6     SecretProvider.GetSecret('PostHogPersonalAPIKey', PersonalAPIKey);
7     SecretProvider.GetSecret('PostHogProjectID', ProjectID);
8     if not PostHogProvider.AddProvider('POSTHOG', PersonalAPIKey, ProjectID) then
9         Error(GetLastErrorText());
10 end;
```



OpenFeature Providers

PostHog Provider

- Good For AppSource Extension
- Feature defined in PostHog and then propagated to all customers.
- Feature state conditions defined in PostHog and then propagated to all customers.


Set 1 Matching users against the criteria

ITManagerOrAdmin

+ Add condition

Roll out to 100 % of users in this set. Will match approximately 0.12% (17 / 14,321) of total users.

DYNAMIC COHORT
Edit View

 ITManagerOrAdmin

Add a description for this cohort

Created 16d ago

PERSONS	LAST CALCULATED
17	6 minutes ago

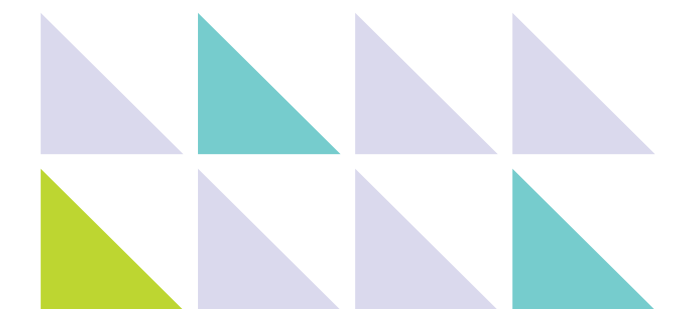
MATCH PERSONS AGAINST ANY CRITERIA

Have the property

profileID exact IT MANAGER

Have the property

profileID exact ADMIN



Things to consider

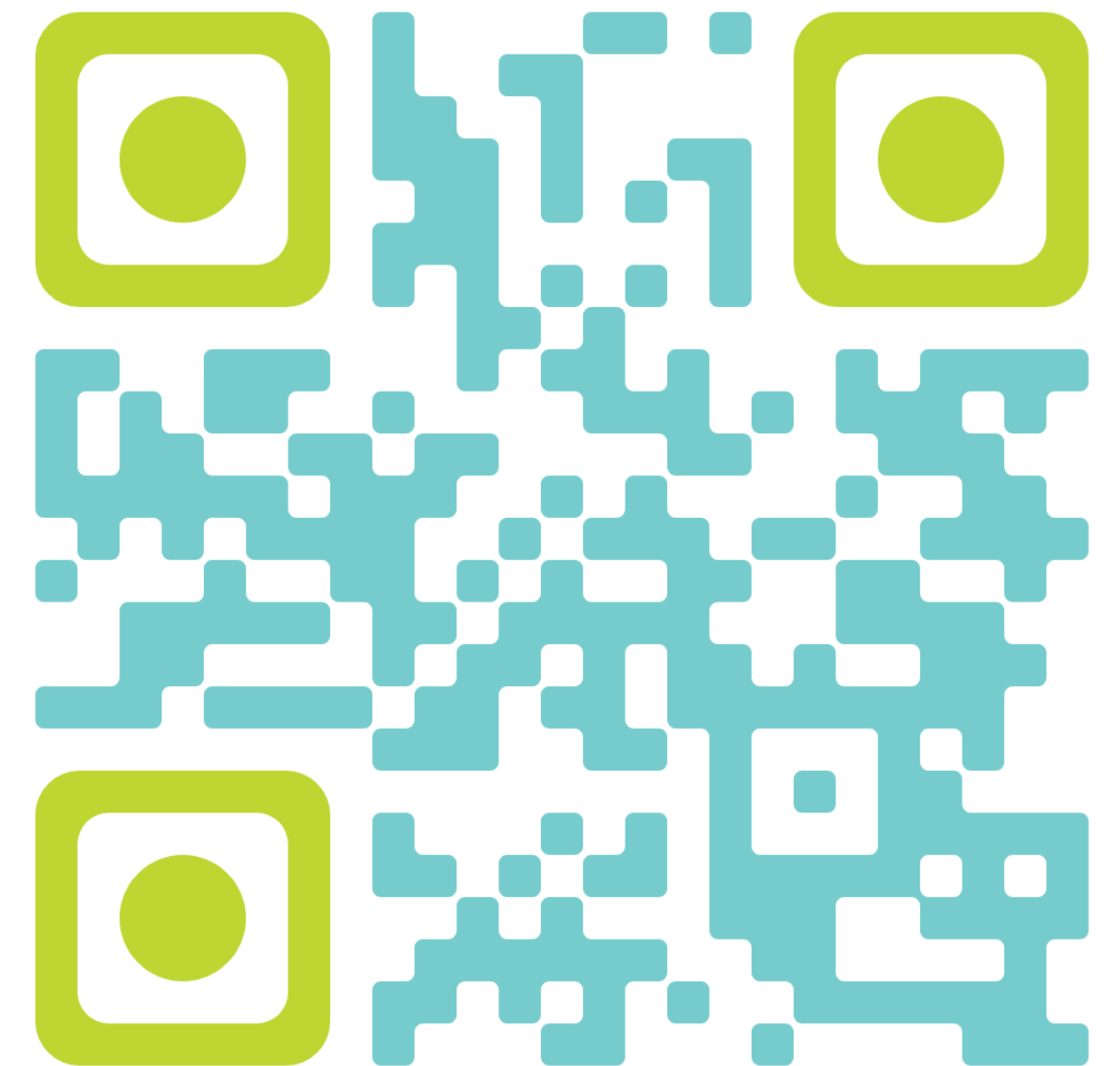
- “Decomposition Overwhelms”
 - warm up your audience with easy-to-sell use cases (User Targeting, On Time)
- Schema Change
 - incremental only! **requires schema design**
 - requires data **upgrade script** at “OnFeatureEnabled” trigger
 - **avoid using user-based conditions** (only company based or always on/off)
- Feature Flag Lifecycle:
 - short-lived - **requires clean-up**
- Feature Flag Dependency:
 - possible, but **STRONGLY NOT RECOMMENDED**
 - backlog **planning with feature isolation in mind**



Takeaways

- Feature Flags approach decouples release from deployment
- Feature Flags approach improves “safeness” and speed of your delivery
- Feature Flags for AL can be handled with open-source tools

365go.in/FF



Q&A

Any Questions?



Thank
You!

