

@bctechdays.com

Thu 22 Jun, 16:00 - 17:30

The System App in examples. How to develop more with less lines of code



Krzysztof
Bialowas



Jesper
Schulz-Wedde



Few words about us

Krzysztof Bialowas

- Developer and System Architect
- Focus on Business Central for more than 12 years
- Training Business Central Partners (ISV and VARs)
- Also helping partners developing solutions
- myNAVblog.com
- #BCBeerinar

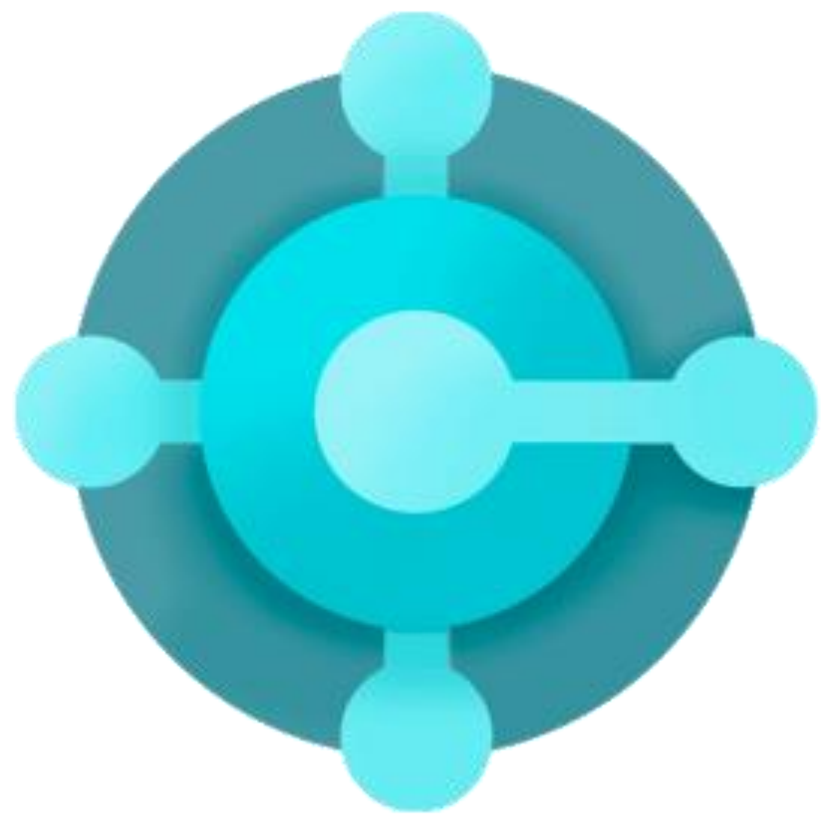


Jesper Schulz-Wedde

- Engineering manager of the Business Central Application Foundation team
- Responsible for Open Development story
- Been on the Business Central team for 15 years
- Trying to bring the community and Microsoft even closer together (e.g. Twitter: @JesperSchulz)

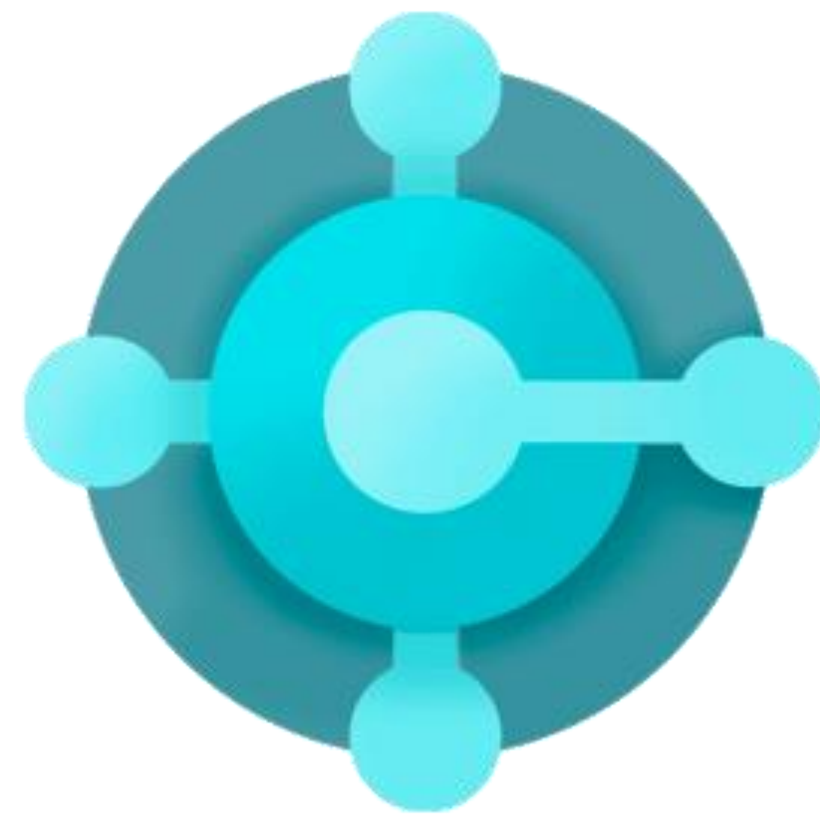


How Business Central is structured (for now)?



System App

Contains libraries and common tools that help Development in AL



Base App

Contains Business Logic for the application such as customers, vendors, posting routines etc.



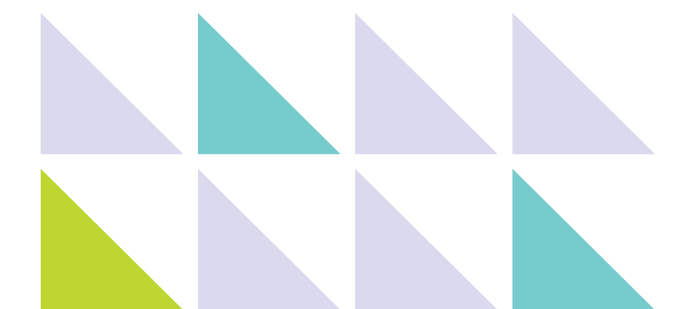
ISV App

Contains additional business logic that can be used in many customers and is not part of Base App



PTE App

Contains solution for exact customer like reports, imports, specific development for customer



Modules like Lego blocks

Azure Storage
Integration

Camera and
Image

Geolocations

Onboarding
Features

Barcodes

Retention
Policies

Custom
Telemetry

oAuth

Base64

SharePoint
Integration

Blob Storage

XMLWriter

ReqEx

...more

GitHub repository: not only SystemApp

Standard APIs

Shopify
Connector

Intrastat

Company Hub

Data Search

Demo data for
Manufacturing

Email
connectors

Test Libraries

Performance
Toolkit



I have a small development task for you

- I want to export XML file with all our customers...
- but it need to be compressed to zip file...
- and it must be sent to azure function...
- but we want to store the copy in some repository (maybe ABS)...
- and we want to send the email with confirmation to the users...
- please make sure it is sent only from production environment...
- and I forgot – we want to send it only every second Monday







Q&A

Any Questions?



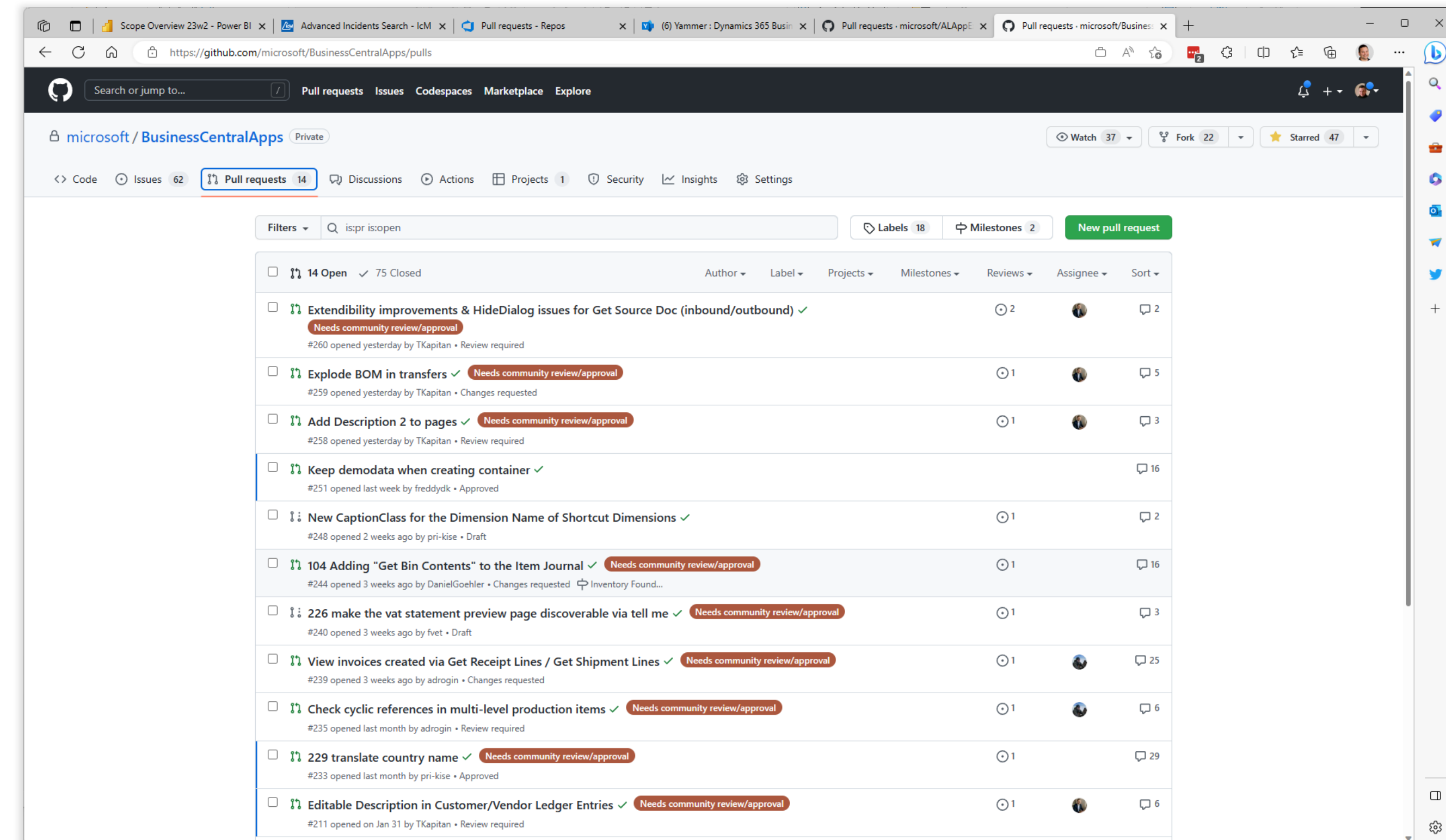
Thank
You!



Disclaimer

Some demos are recorded for optimal viewer experience and hence cannot be adjusted in size. Please come to the front, if you are having difficulties seeing what's on the screen.

And now: **Let's jump into it!**



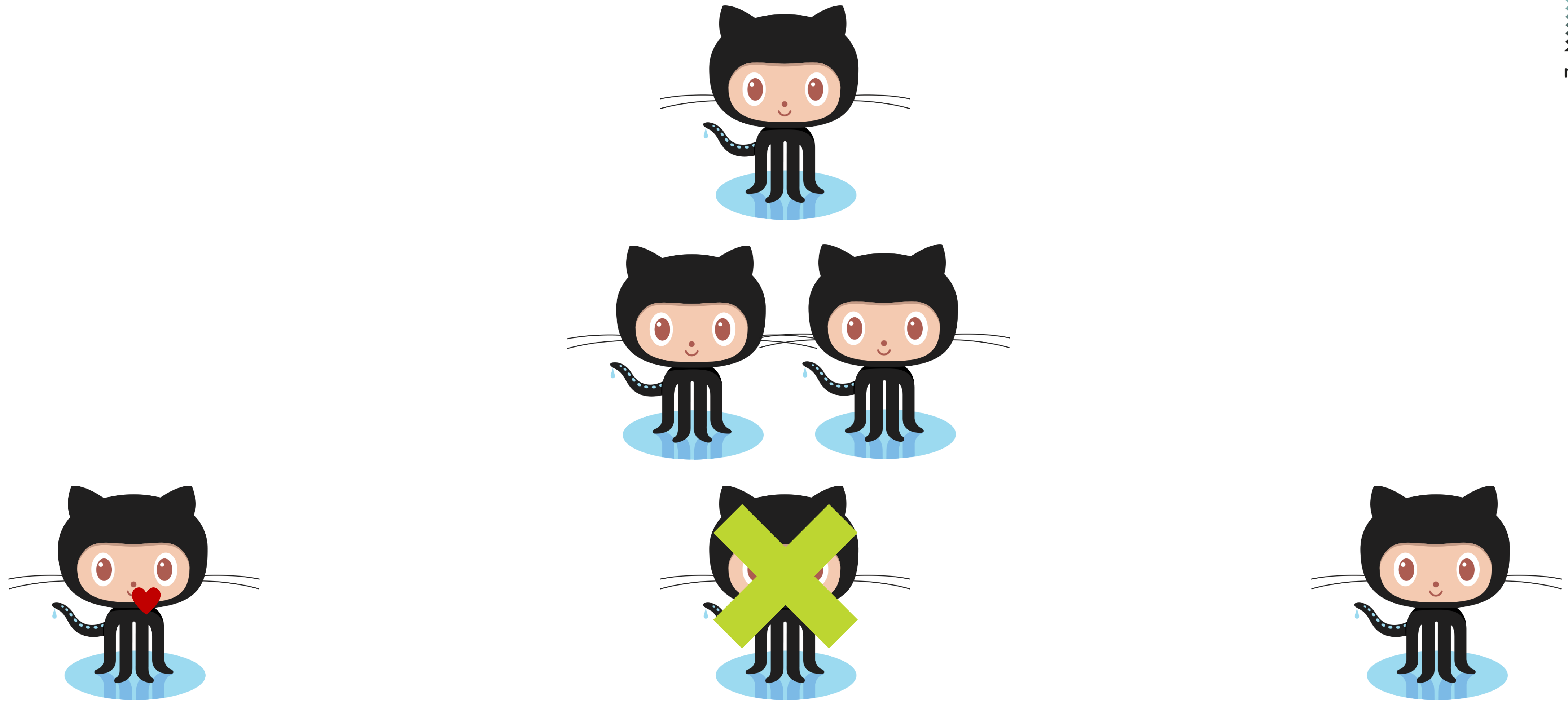
Agenda

- Why have an Open Development model?
- The contribution process:
 - Get your contribution suggestion approved
 - Get your development environment setup
 - Create your PR and get it to the finish line
- What's next? Which changes are to come?
 - Going Full Open Source
 - Introducing the Business Foundation



Why have an open development model?





Together we win!



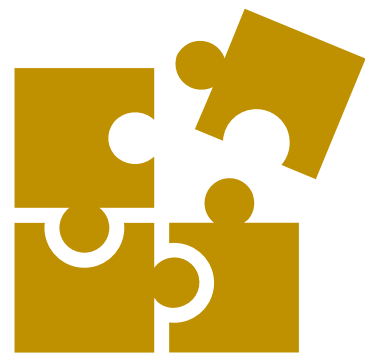
Why have an open development model?



Accelerate Product Growth



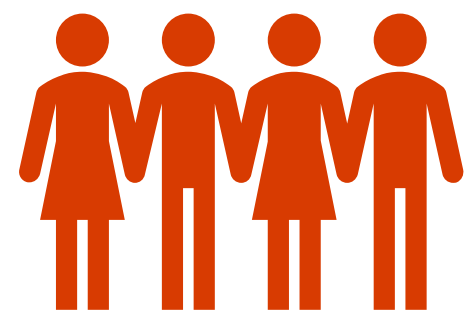
Improve Product Quality & Auditability



Enhance Extensibility, Customizability & Compatibility
Focus on Last Mile



strong foundation
Foster Collaboration



Develop Community

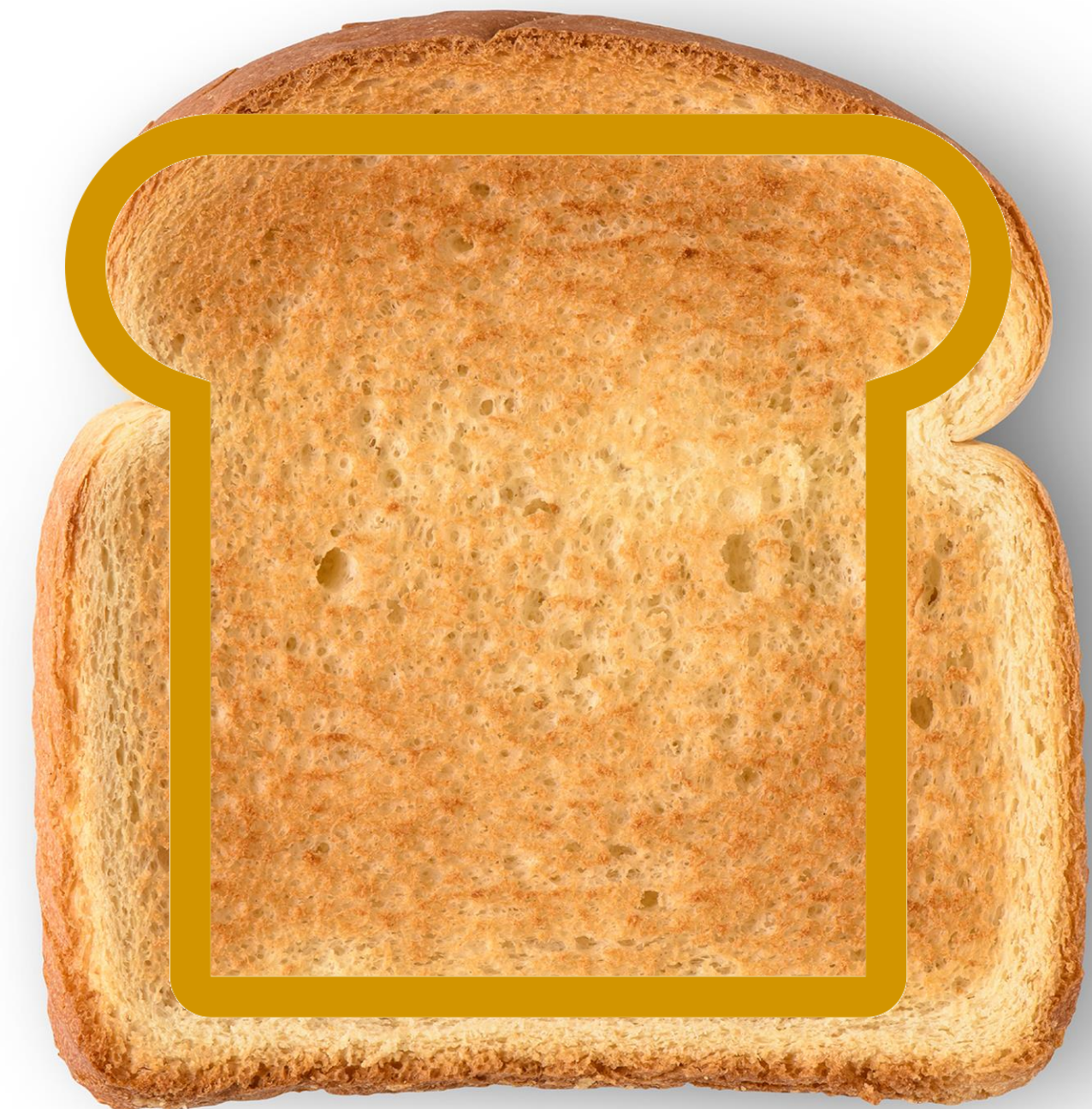


The contribution process

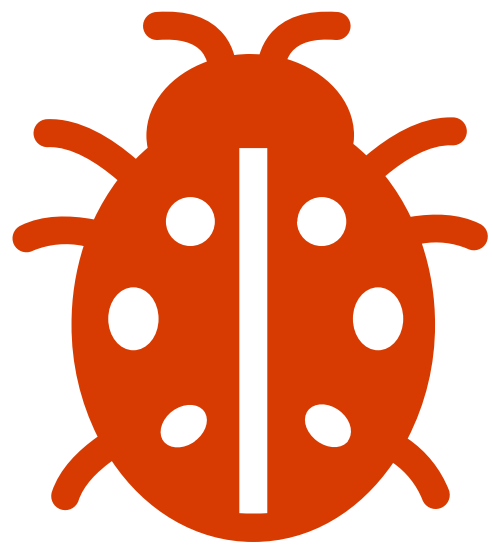
Phase I: Get your idea approved for contribution



How do we work internally at Microsoft?



How do we work internally at Microsoft?



Customer reported product defects (support cases, repair items and hotfix requests)

Procedural issues (e.g., blocking work processes)

Data issues (e.g., generating faulty data)

Partner reported product defects

Papercuts (typically fixed by themselves in a code customization model)

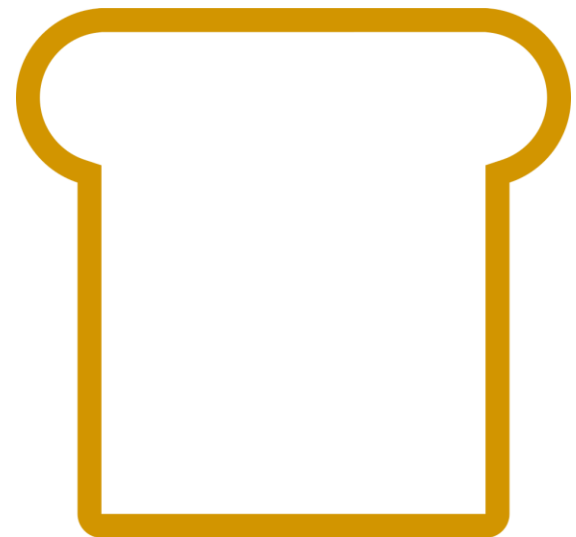
Design change requests (added to prioritized backlog)

Collaboration requests (helping partner resolve an issue on their end)

Extensibility requests and issues (currently processed by separate team)

Internally reported product defects

Come in all shapes and sizes



Part of internal epics (e.g., fundamentals, onboarding, manufacturing)

Large stories which improve the product all up

Partner reported ideas (aka.ms/bcideas)

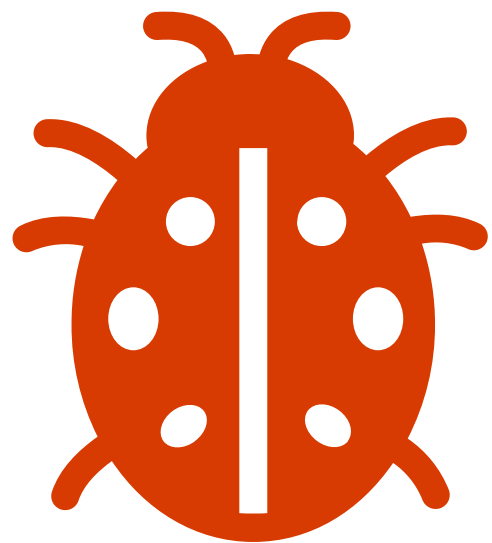
Ideas of various nature and various sizes

!!! EXIT CRITERIA !!!



...And what are contribution candidates?

GitHub Issue



Customer reported product defects (support cases, repair items and hotfix requests)

Procedural issues (e.g., blocking work processes)

Data issues (e.g., generating faulty data)

Partner reported product defects

Papercuts (typically fixed by themselves in a code customization model)

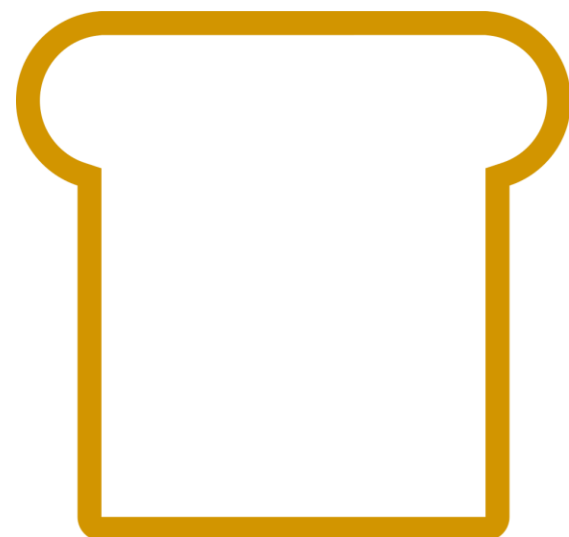
Design change requests (added to product backlog)

Collaboration requests (helping partner solve an issue on their end)

Extensibility requests and issues (currently processed by separate team)

Internally reported product defects

Come in all shapes and sizes

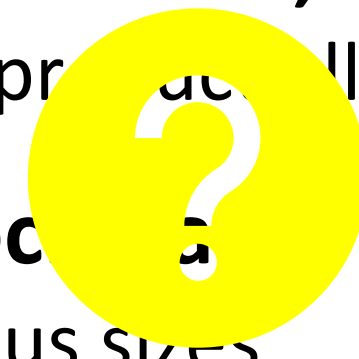


Part of internal epics (e.g., fundamentals, onboarding, manufacturing)

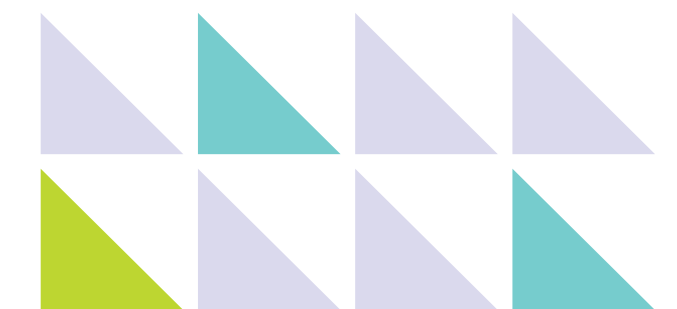
Large stories which improve the product all up

Partner reported ideas (aka.ms/bc-ideas)

Ideas of various nature and various sizes



BC Idea



What will product managers approve?

Always

- Papercuts
 - Missing fields on pages
 - Missing fields in data sets
 - Typos or missing tooltips
 - Missing actions found elsewhere
 - Alignment of experience
 - Improvements to tests
- New horizontal building blocks
- Bug-fixes without production impact

Sometimes

- Changes to existing business logic
- Very large and complex contributions

Never

- Product Defects with customer impact
- Work on localizations
- Currently: Event Requests



Demo 1: Create your GitHub issue

And get it “ready for implementation”



Business Manager - Dynamics 365

https://businesscentral.dynamics.com

Dynamics 365 Business Central

Environment: Production

Search

Notifications

Flags

Settings

Help

User Profile

Fourth Coffee

Finance

Cash Management

Sales

Purchasing

Shopify

Customers

Vendors

Items

Bank Accounts

Chart of Accounts

Insight from the last three months

The best-selling item was Guest Section 1 with 47 units sold

Actions

- + Sales Quote
- + Sales Order
- + Sales Invoice
- + Purchase Quote
- + Purchase Order
- + Purchase Invoice
- > Find entries...
- > Search in data...
- > New
- > Payments
- > Reports
- Excel Reports

Get started: Here are a few things you can try out

Show demo tours

Activities

Sales This Month

- \$1,120
- > See more

Overdue Sales Invoice Amount

- \$70,037
- > See more

Overdue Purch. Invoice Amount

- \$49,872
- > See more

Sales Invoices Predicted Overdue

- 0
- > See more

Ongoing Sales

- Sales Quotes
 - 14
 - >
- Sales Orders
 - 60
 - >
- Sales Invoices
 - 11
 - >

Ongoing Purchases

- Purchase Orders
 - 19
 - >
- Ongoing Purch. Invoices
 - 5
 - >
- Purch. Invoices Next Week
 - 0
 - >

Payments

- Unprocessed Payments
 - 1
 - >
- Average Collection Days
 - 0.0
- Outstanding V. Invoices
 - 14
 - >

Scan documents

- Create Incoming Invoices from Camera

Incoming Documents

The contribution process

Phase II: Get your development environment set up

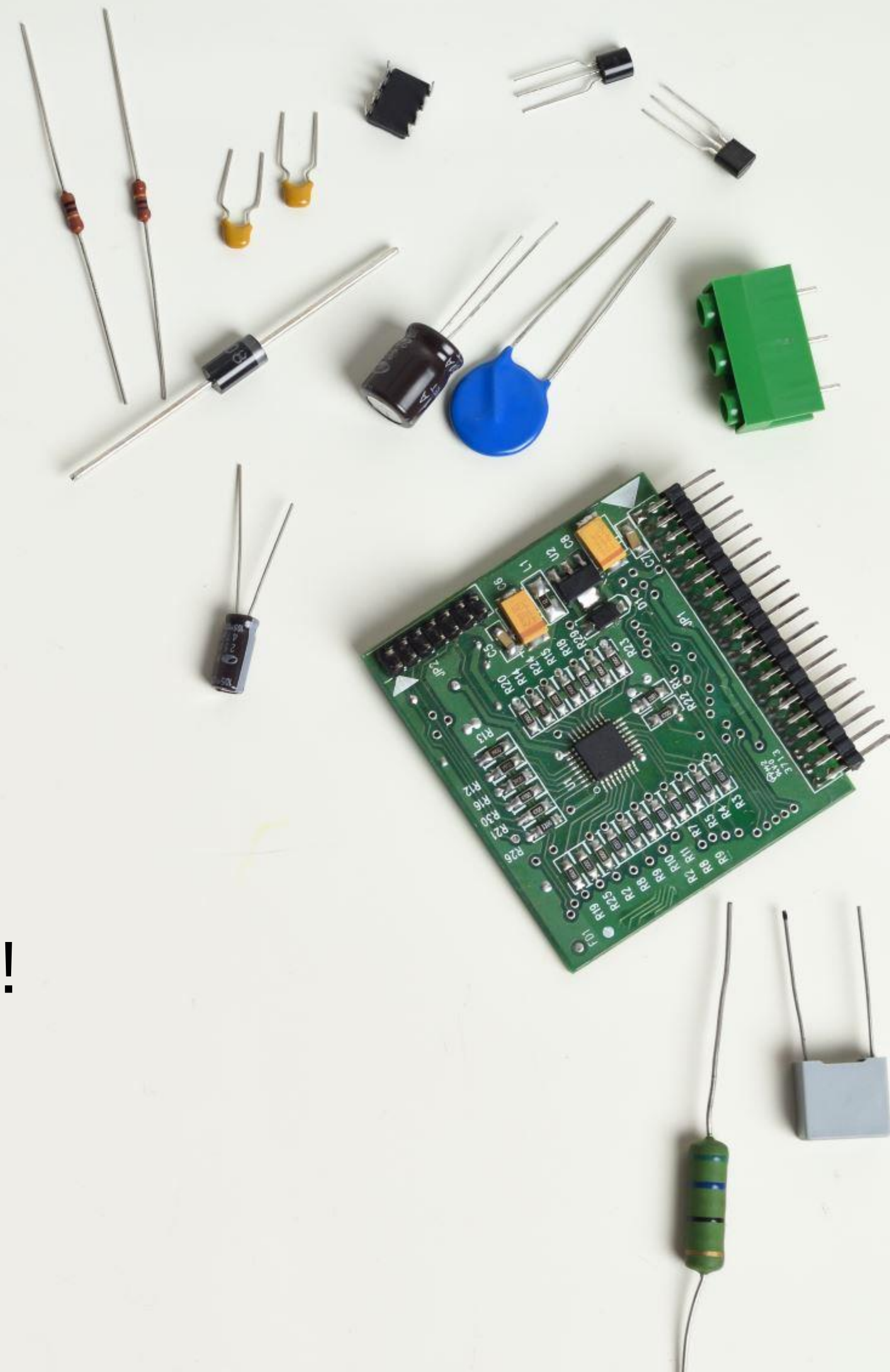


It's easy – really...

Once you know how!

So, let's take 10 minutes to walk you through the setup process of a Business Central development machine.

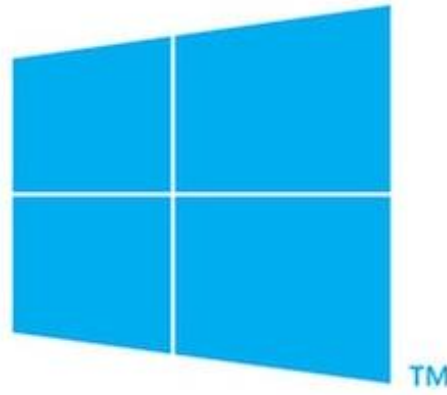
We'll be with you every single step of the way!



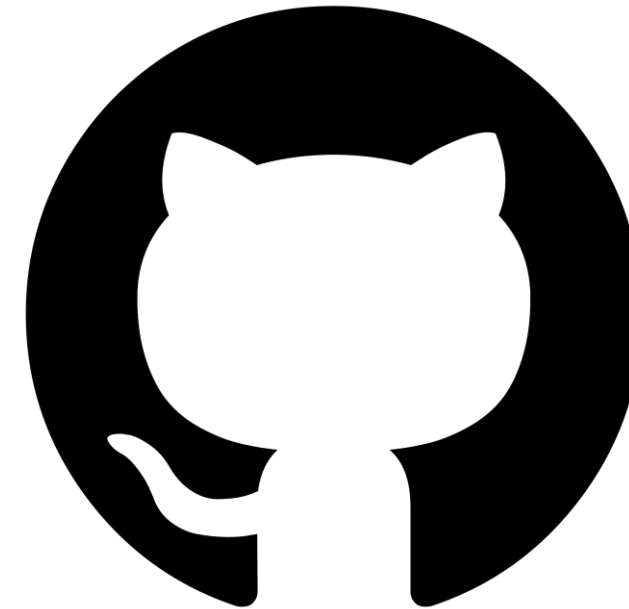
First things first: You need a PC



Requirements for getting started



Windows®

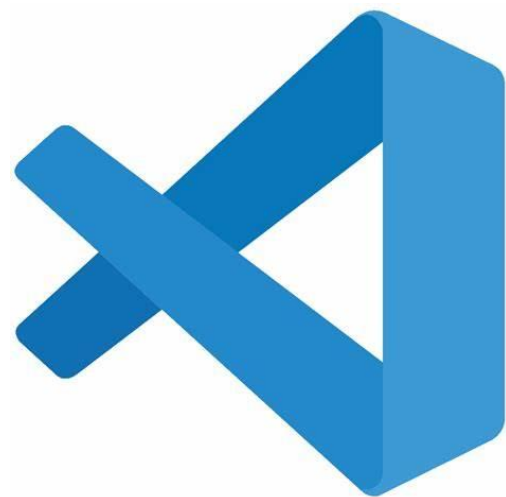


GitHub Account



git

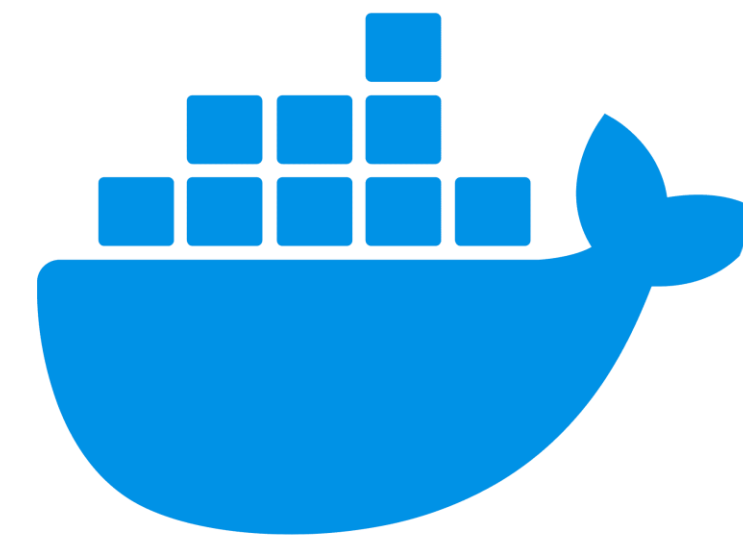
Basic git understanding



Visual Studio
Code



AL Language
Extension



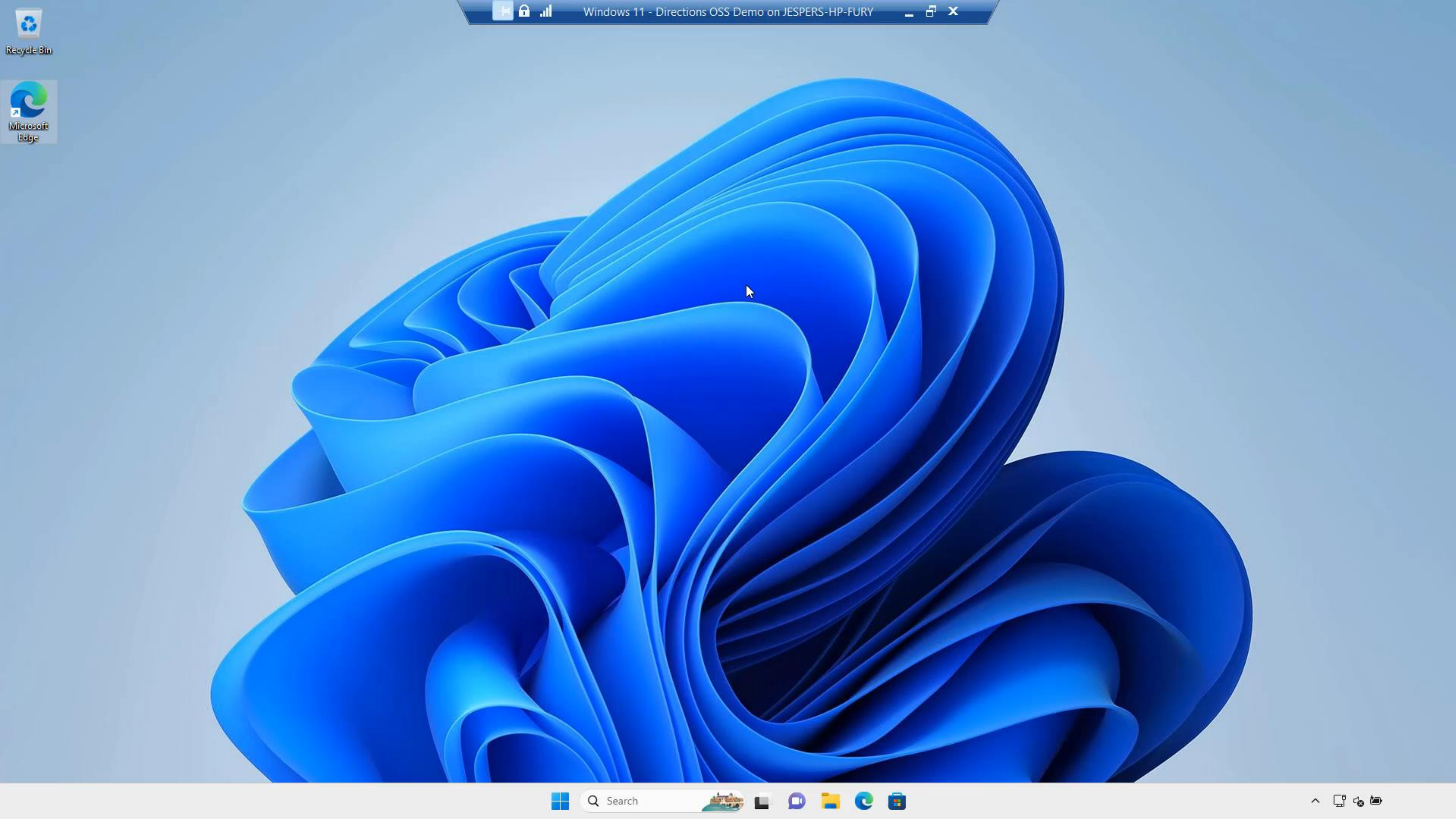
docker®



Demo 2: Get your development environment set up

From 0 to contribution hero in 30 minutes





Search



**What other
settings should
one make?**

Enable RAD!



The contribution process

Phase III: Create a PR and get it released



Demo 3: Create your Pull Request

And get it released





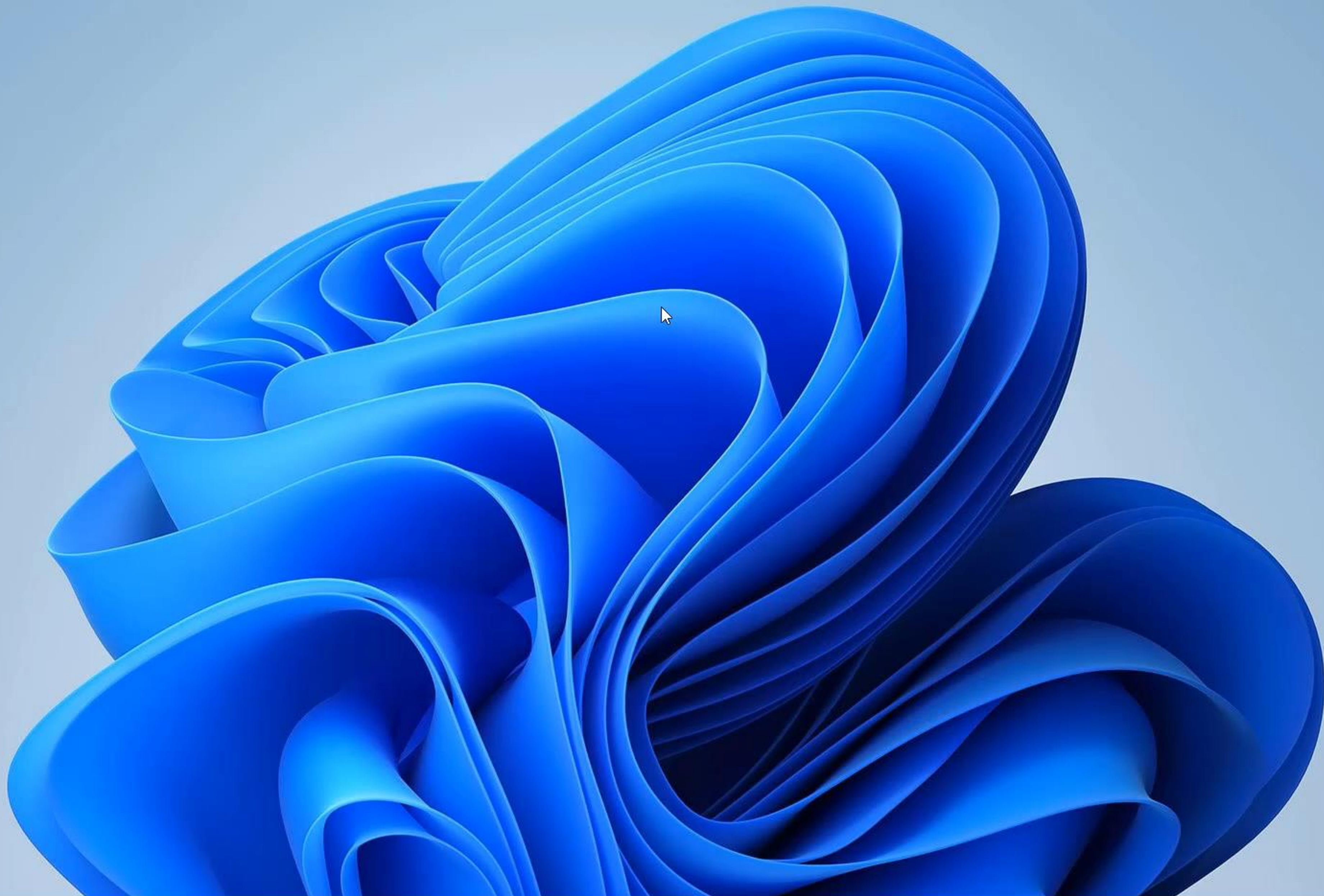
Recycle Bin



Microsoft
Edge



Docker
Desktop



Q Search



What's Next?



What's next?

The System Application is moving!

Why?

A fresh start...

Why?

The Business Central application team is finally moving development to GitHub as well!

Why?

Because only then will we reap the full benefits of open development!



Where?

We won't be going far: <https://github.com/microsoft/BCApps>

When?

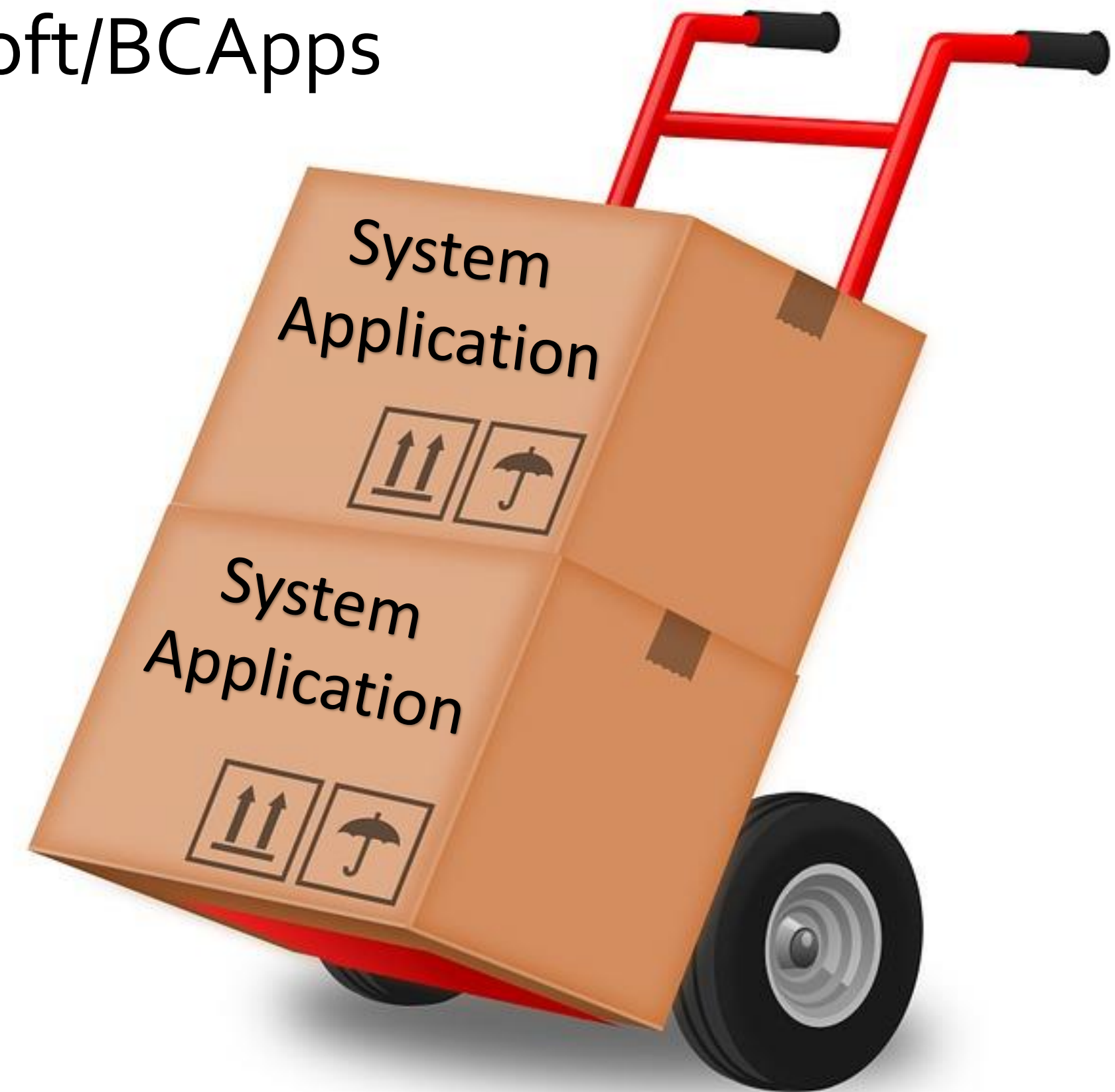
When our CI/CD pipeline is ready.

When we've established the new processes.

We're getting there!

How?

From one day to another.



Is that it?



What's Next?

The System Application will be getting a new upstairs neighbor!

Why?

We want to continue our efforts to componentize the business application platform. The System Application was only ever meant for non-ERP specific, platform-like capabilities.

What?

We will be extracting the very first module with business purpose from the Base Application, thereby establishing the Business Foundation

How?

That requires more than one bullet to explain 😊



How?

Non-breaking to apps above the App Foundation!

Extensions

Starting out slowly:
No. Series module only!

Extensions

Application

Base Application

Functionality used by **ALL** or **NONE** of the application domains

Implemented after same principals as the System Application:

- Better, cleaner APIs
- Well documented
- Fully extensible
- Improved capabilities
- No local versions

System Application



What's Next for the Business Foundation?

Dimensions

Currency

Location

Company
Info

Unit of
Measure

Business
Unit

Source
Codes

Reason
Codes

(Accounting)
Periods

...

