BC/NAV TECH DAYS 2022

mibuso.com

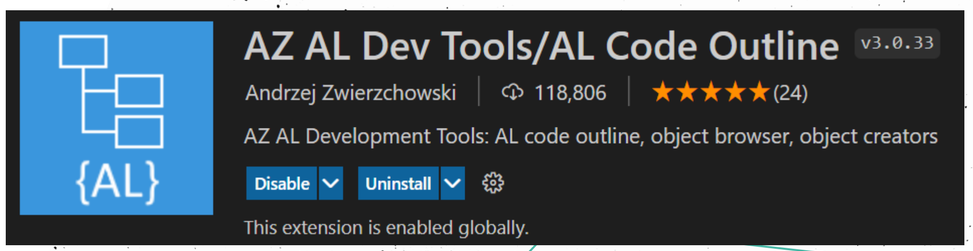
Improve quality and readability of your code

Andrzej Zwierzchowski
The NAV People

www.bctechdays.com

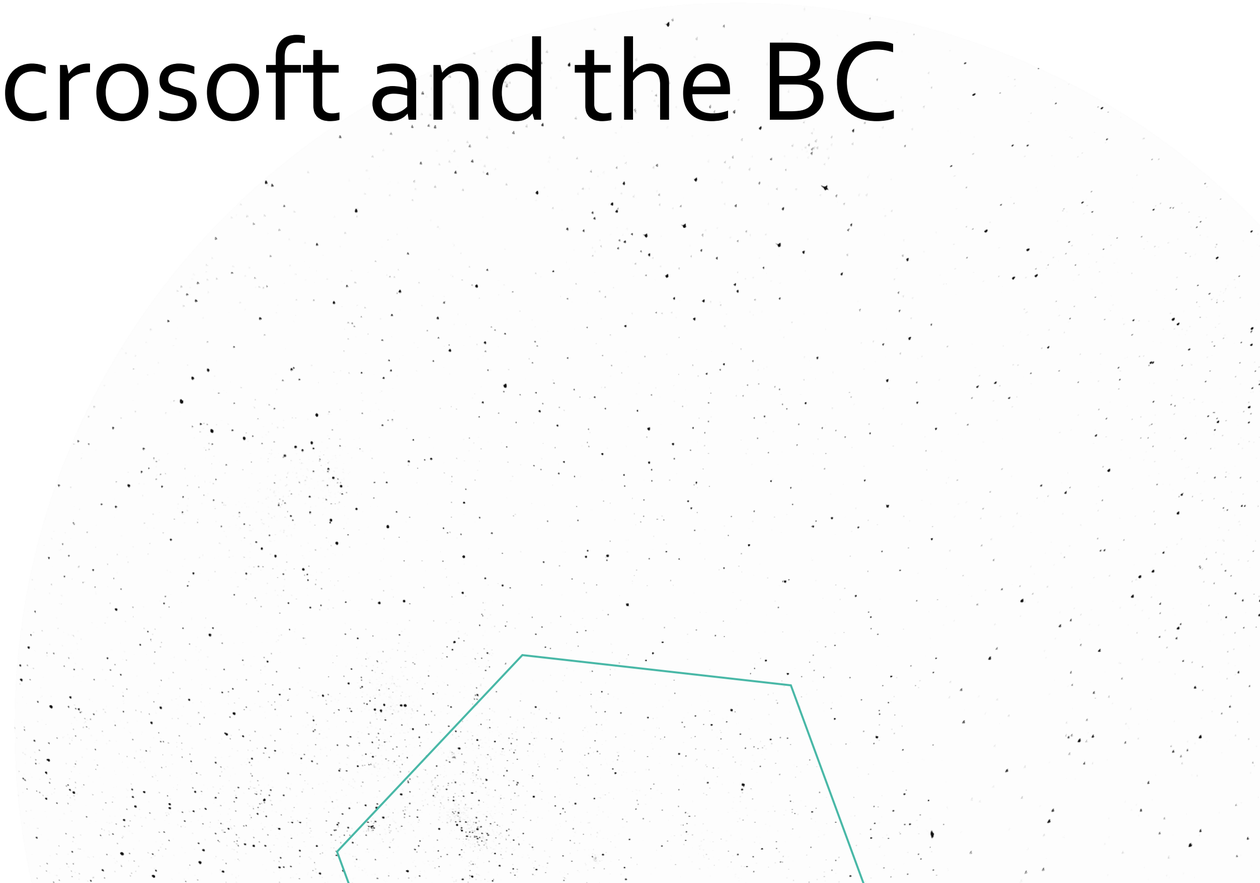
About me

- Development Standards Lead at The Nav People
- C#, C/AL, AL and (sometimes) JS developer
- Author of "AZ AL Dev Tools" extension
- Twitter: https://twitter.com/anzwdev
- Blog: https://anzwdev.wordpress.com/



Session subject

Improve the quality and readability of your code using tools from Microsoft and the BC Community



Quality and readability

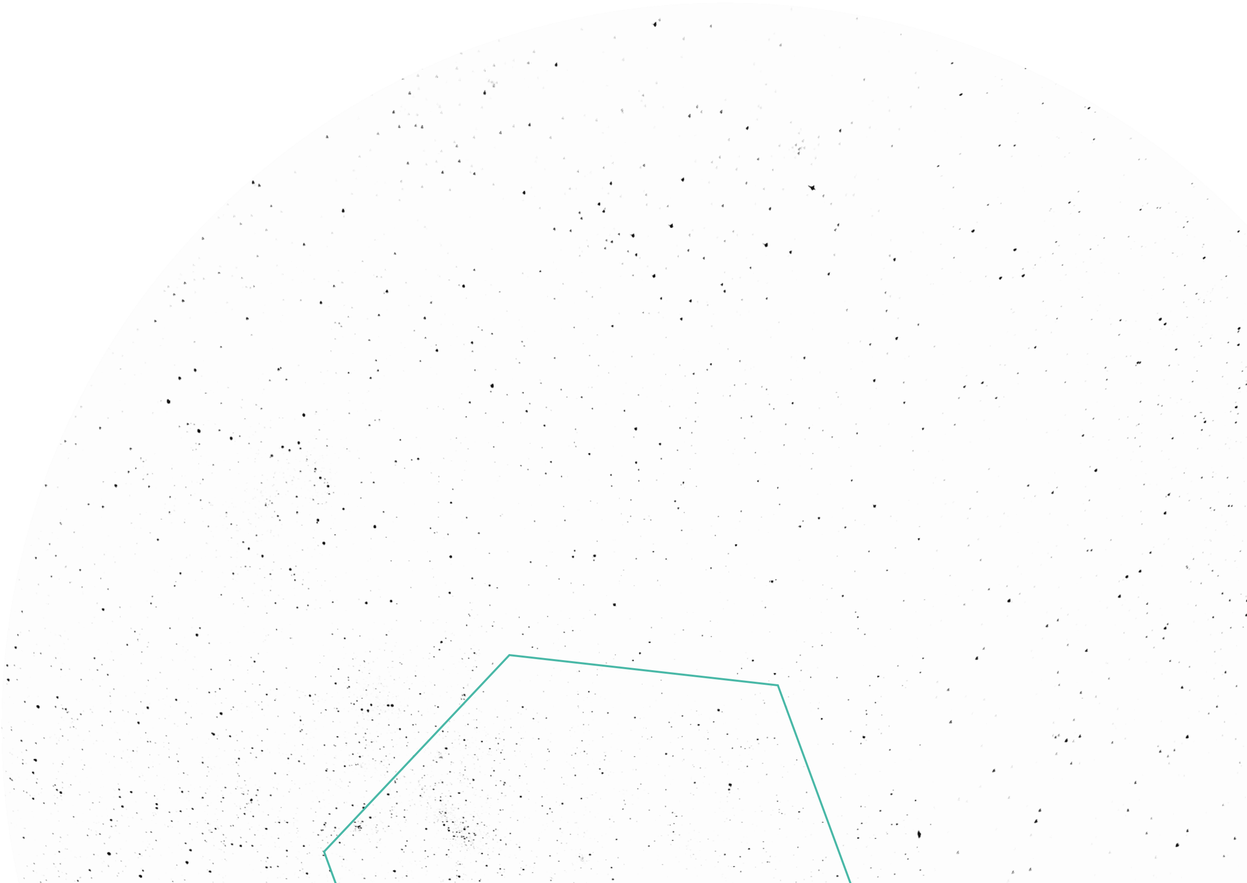
- Source of information for other developers
- Reading to writing ratio is usually greater than 10:1
- We are part of a team
- Anybody should be able to fix or extend our code
- Bad code slows us down

What can help us

- Source control
- Development guidelines and patterns

  https://alguidelines.dev

  Community run and Microsoft endorsed.
- Code Analyzers
  - Microsoft
  - BusinessCentral.LinterCop by Stefan Maron
- VS Code Extensions
  - Snippets
  - Commands
  - Code Actions
  - Additional information (VS Code panels)



Setup

- User settings for VS Code

```
{
    "workbench.colorTheme": "Default Dark+",
    "workbench.iconTheme": "vscode-icons",
    "editor.fontSize": 18
}
```

- Workspace settings for the solution

.vscode > {} settings.json > ...

```
{
    "al.enableCodeActions": true,
    "al.enableCodeAnalysis": true,
    "al.backgroundCodeAnalysis": true,
    "al.codeAnalyzers": [
        "${CodeCop}"
    ],
    "CRS.FileNamePattern": "<ObjectNameShort>.<Obje
    "CRS.FileNamePatternExtensions": "<ObjectNameSh
    "CRS.FileNamePatternPageCustomizations": "<Obje
    "CRS.OnSaveAlFileAction": "Reorganize"
```

- Keep workspace settings in the source control

  .vscode\settings.json

  Don’t add it to the .gitignore file

Magic numbers

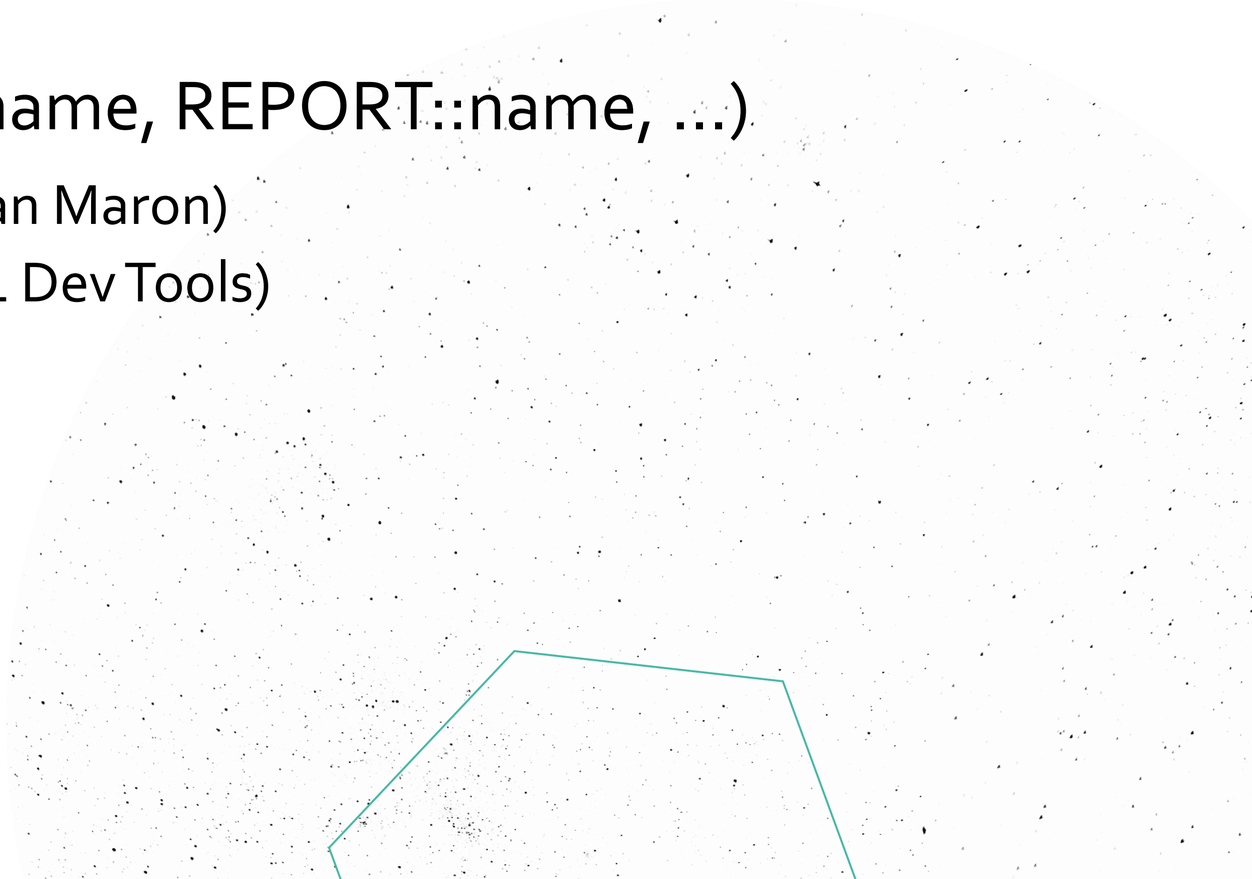
- Values with unexplained meaning
- It breaks one of the oldest rules of programming (Robert C Martin – “Clean Code”)
- Use enums

```
procedure ProcessDocument(ActionType: Integer; var SalesHeader: Record "Sales Header")
begin
    case ActionType of
        1:
            SendEmail(SalesHeader);
        2:
            PostDocument(SalesHeader);
        3:
            PrintDocument(SalesHeader);
    end;
end;
```

Hardcoded Object Ids

BC/NAV TECH DAYS 2022
mibuso.com

- Do not hardcode object ID
- Assign Id and forget it
- Use system enum types (DATABASE::name, REPORT::name, ...)
  - Detect using BusinessCentral.LinterCop (Stefan Maron)
  - Fix with "Convert Object Ids to Names" (AZ AL Dev Tools)
- Reserve IDs
  - Your own solution
  - AL Object ID Ninja (Vjeko)



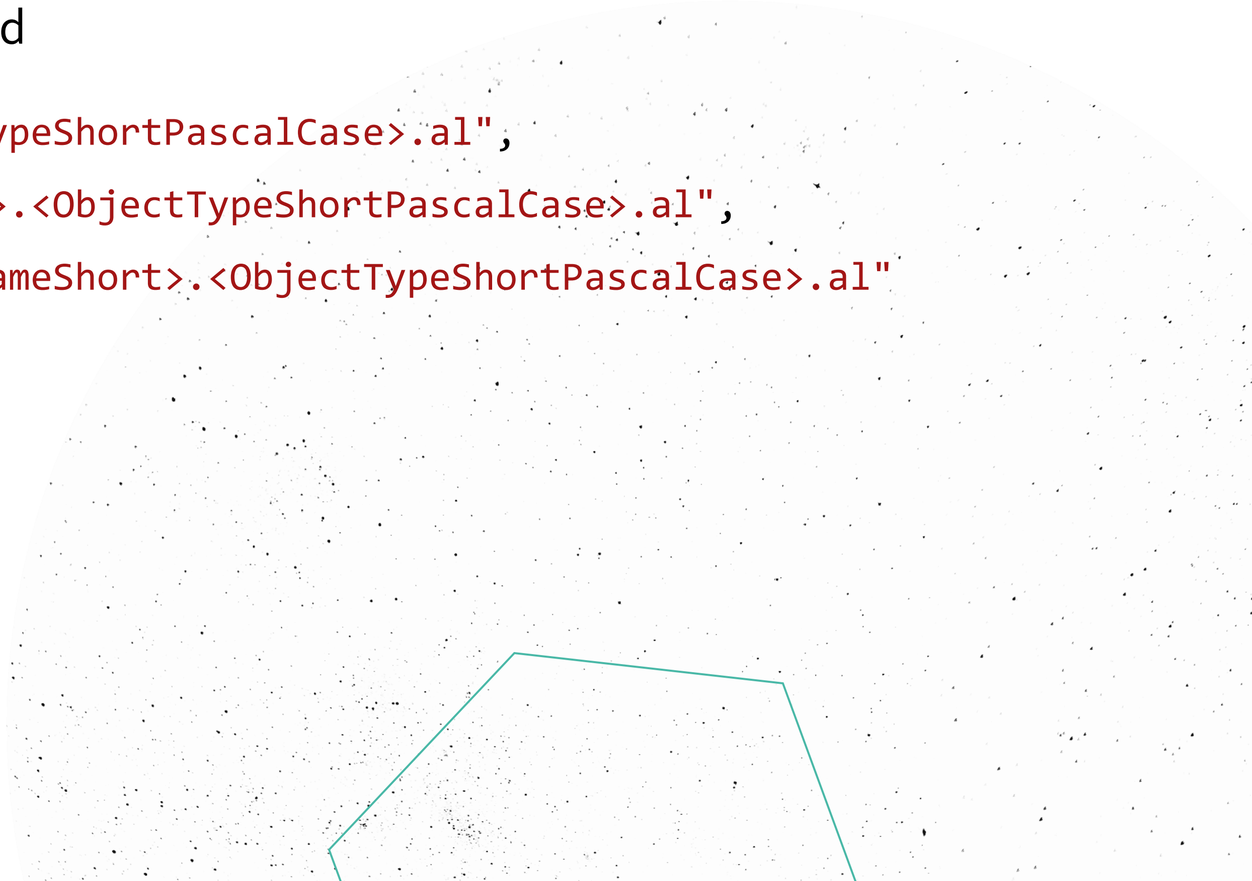
Hardcoded Object Ids

BC/NAV TECH DAYS 2022
mibuso.com

- Do not keep object IDs inside file names
  - Use Waldo's CRS AL Language Extension

    “CRS: Configure Best-practice File Naming” command

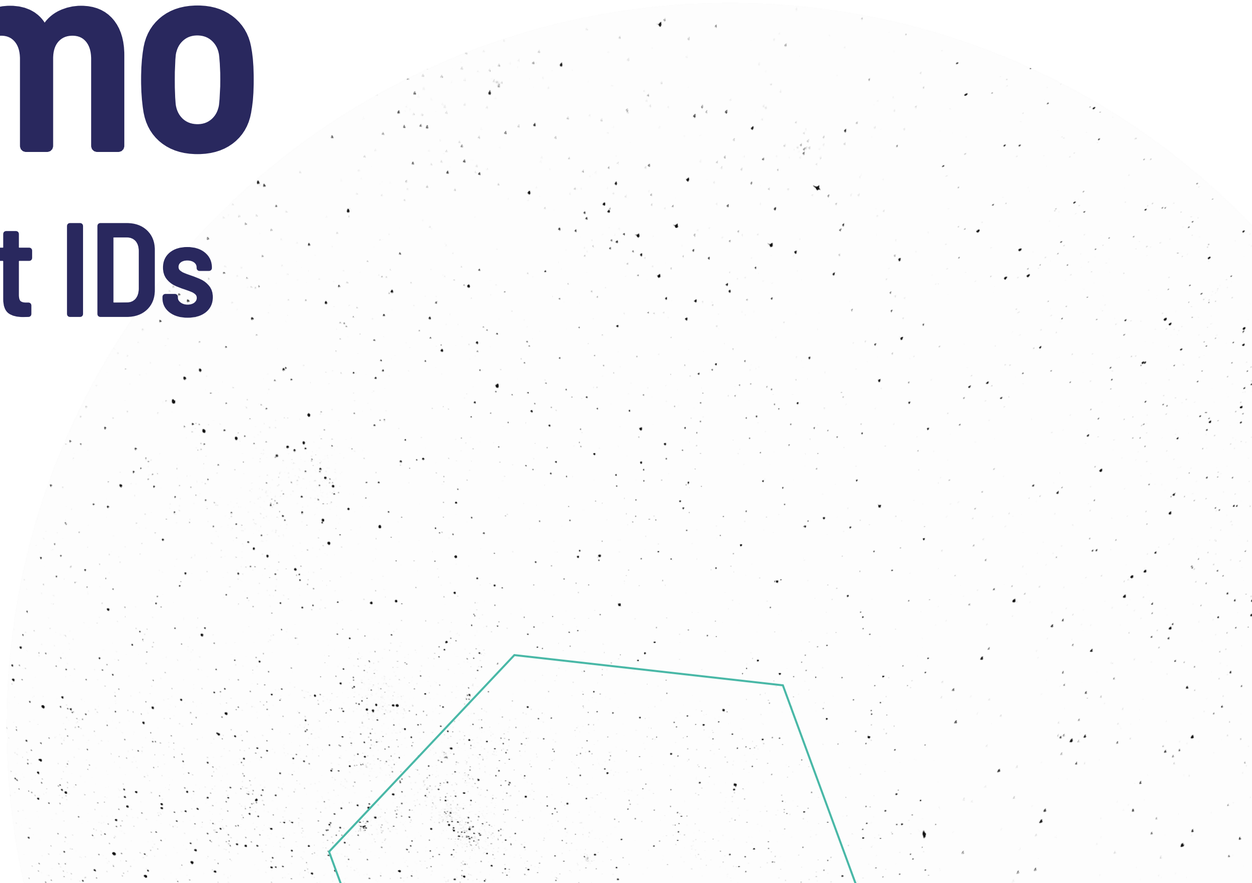
```
"CRS.FileNamePattern": "<ObjectNameShort>.<ObjectTypeShortPascalCase>.al",
"CRS.FileNamePatternExtensions": "<ObjectNameShort>.<ObjectTypeShortPascalCase>.al",
"CRS.FileNamePatternPageCustomizations": "<ObjectNameShort>.<ObjectTypeShortPascalCase>.al"
```



BC/NAV TECH DAYS 2022
mibuso.com

Demo

Object IDs



Procedures

- Should be small
- Should do one thing
  - Robert C Martin – One thing means that you cannot extract another procedure
  - Use “Extract Procedure” from “AL Code Actions”
- Name should explain functionality
  - Rename of you have a better name (F2 in VS Code)

BC/NAV TECH DAYS 2022
mibuso.com

Demo

Extract Procedure

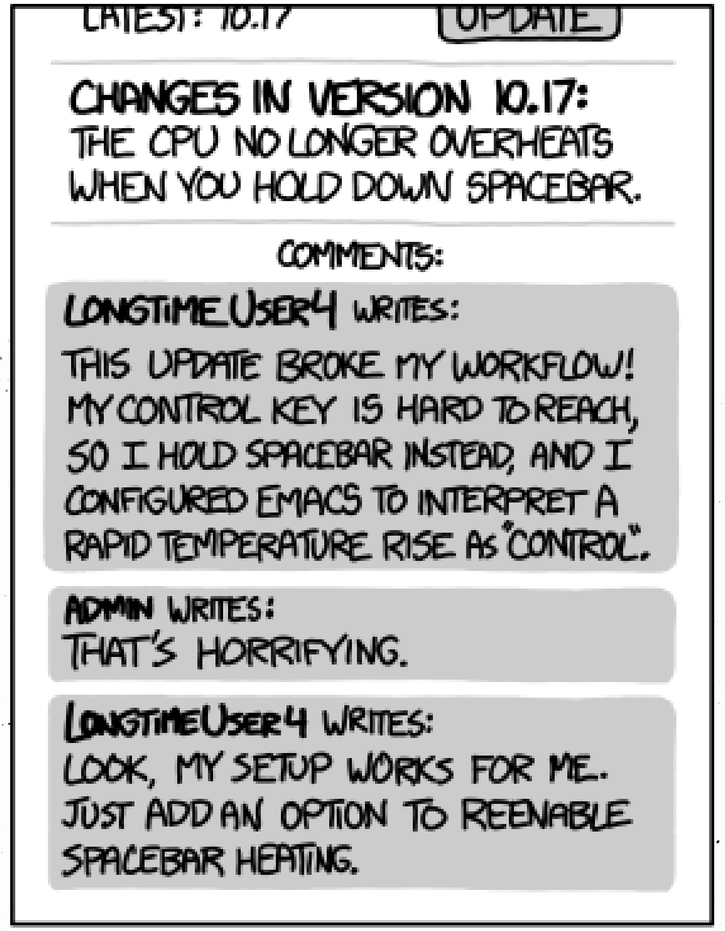
Procedures – A problem

Hyrum's Law

With a sufficient number of users of an API,
it does not matter what you promise in the contract
all observable behaviours of your system
will be depended on by somebody.

https://www.hyrumslaw.com/

It makes refactoring harder



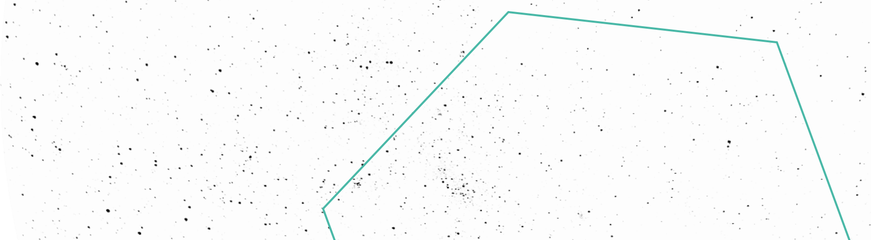
EVERY CHANGE BREAKS SOMEONE'S WORKFLOW.

https://xkcd.com/1172/

Procedures – Solution

Keep implementation private

- Use local procedures
- Façade pattern - https://alguidelines.dev/docs/patterns/facade-pattern/
- Consider using interfaces
  - "Interface Wizard" and "Create Interface" code action from "AZ AL Dev Tools" can help
- Design your event publishers, don't create them for every procedure

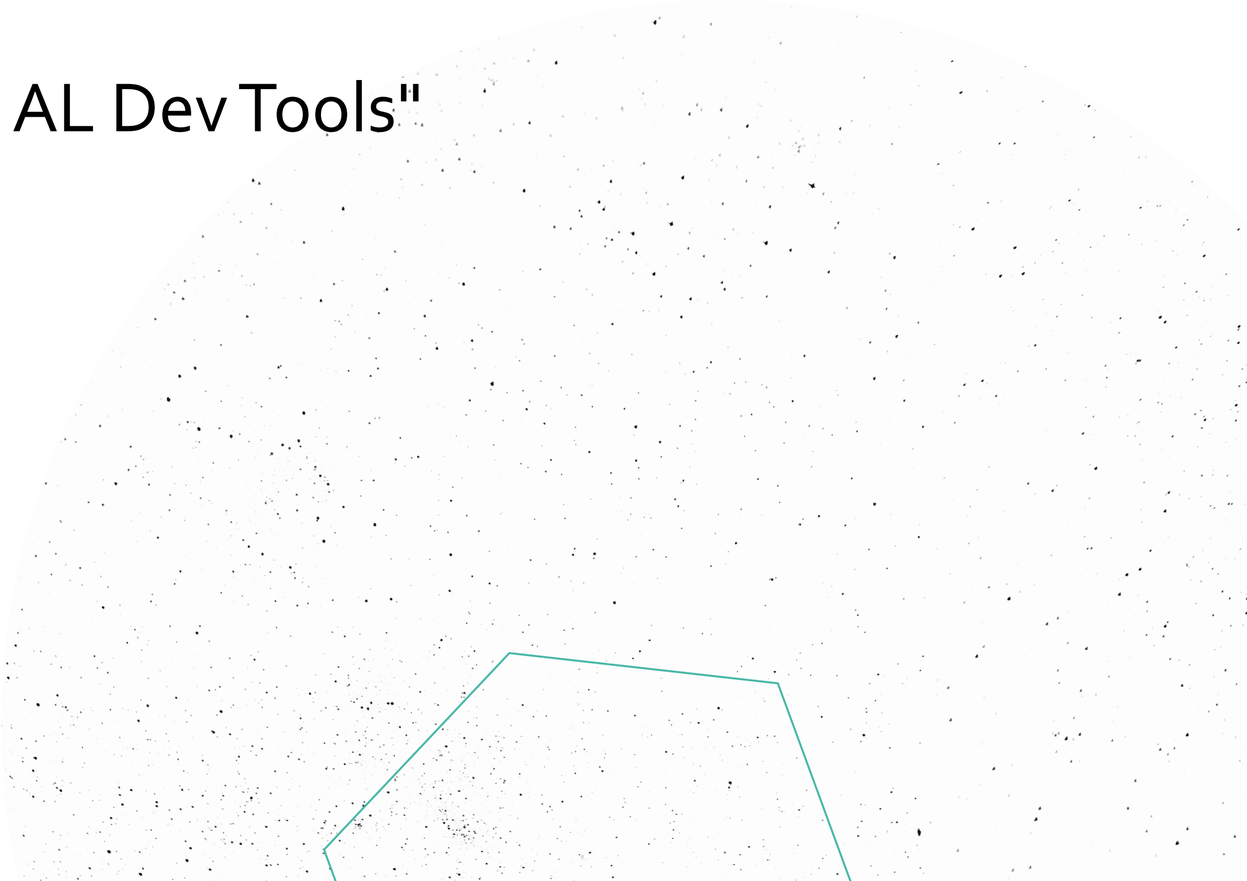


Demo

Create interface

Avoid duplication

- Move duplicated code to a procedure
- Group procedures in codeunits if they are related
- Use "Find Duplicate AL Code" from "AZ AL Dev Tools"



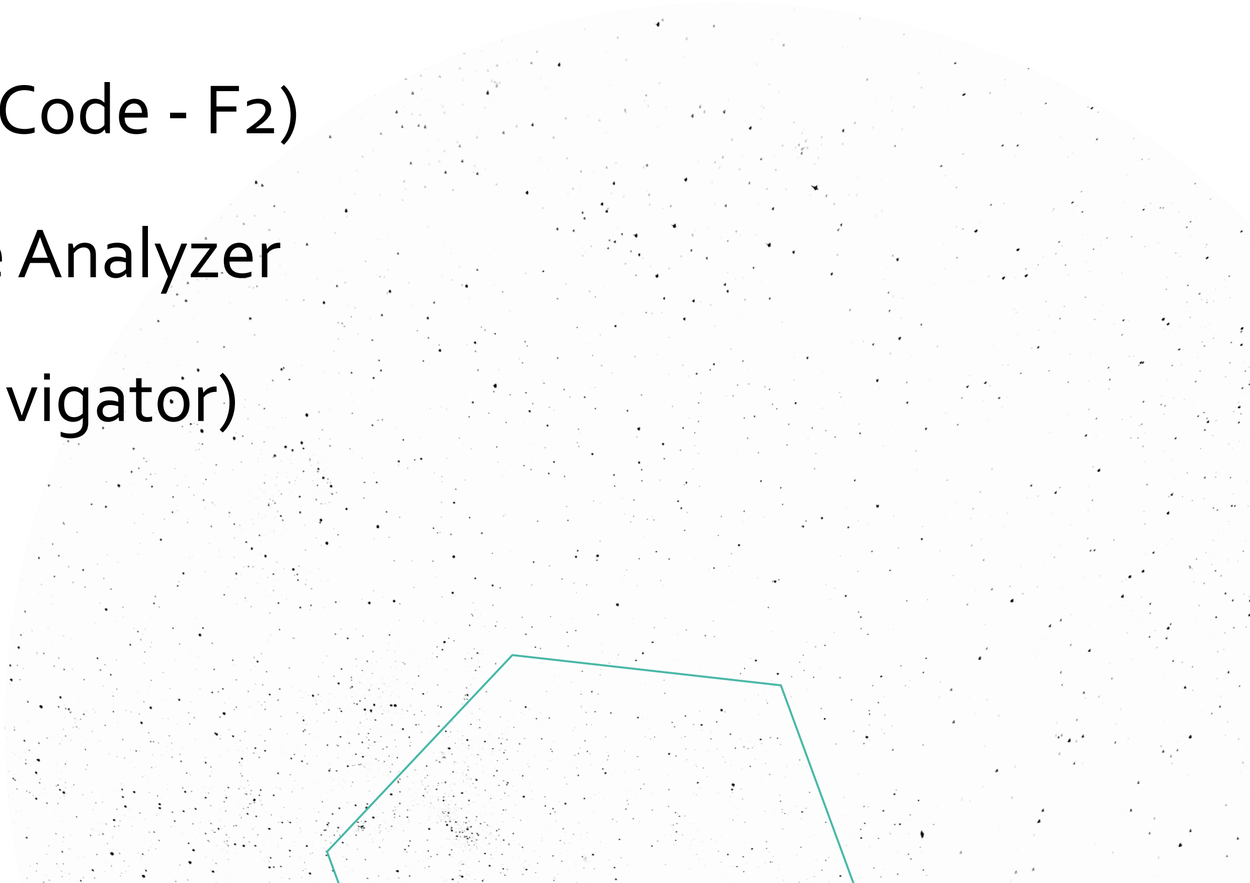
BC/NAV TECH DAYS 2022
mibuso.com

Demo

Duplicate code

Variable names

- Names should reveal intention
- Avoid names that look similar
- Rename if you have a better name (VS Code - F2)
- Detect problems with “CodeCop” Code Analyzer
- Use "Add variable" code actions (AL Navigator)
- Snippets from AL Variable Helper
- Code completion (AZ AL Dev Tools)



BC/NAV TECH DAYS 2022
mibuso.com

Demo

Create variables
Detect variable names issues

Comments

- “My experience suggests that other human beings don't read comments and compilers don't read comments, so who you writing them for.”

  Kevlin Henney - Seven Ineffective Coding Habits of Many Programmers
  https://www.youtube.com/watch?v=ZsHMHukIlJY

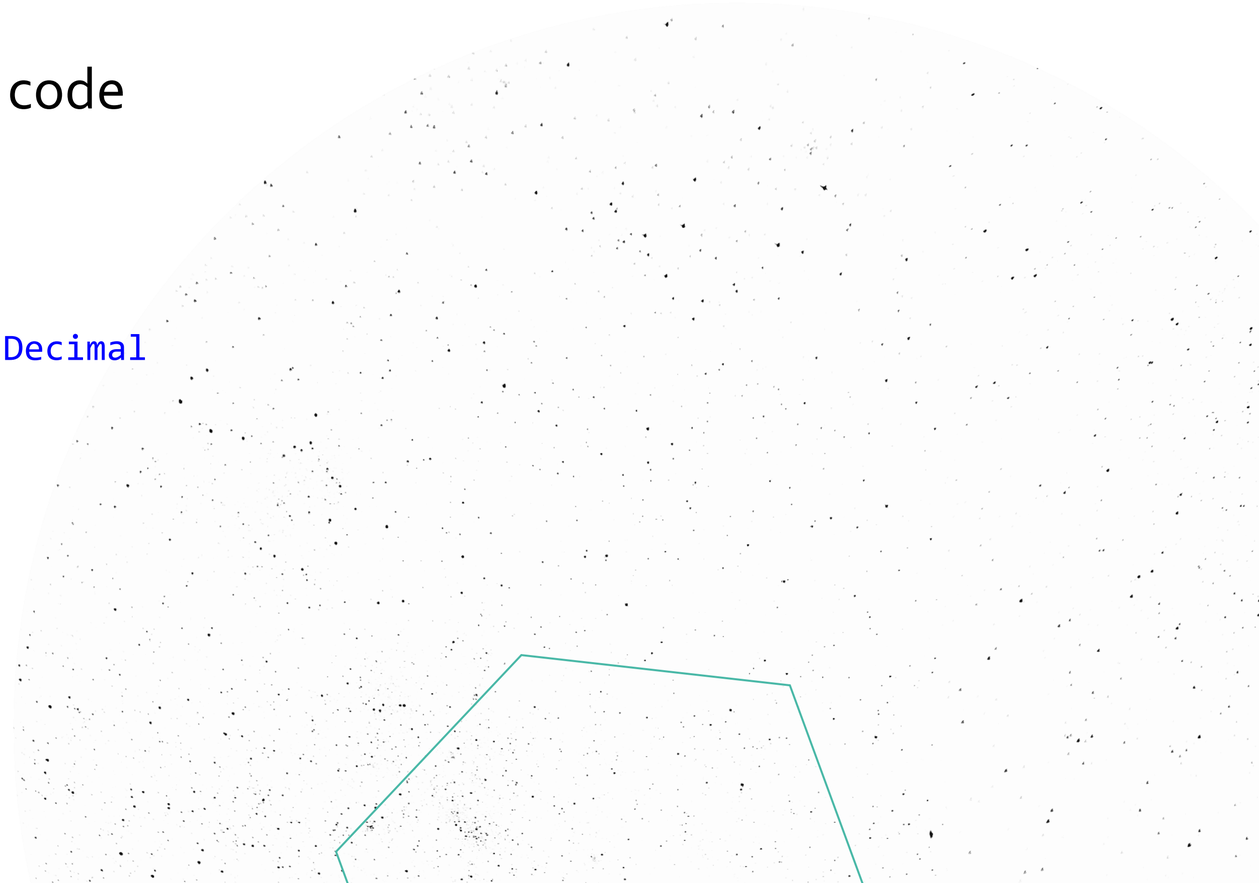
- “A common fallacy is to assume authors of incomprehensible code will somehow be able to express themselves lucidly and clearly in comments.”

  https://twitter.com/KevlinHenney/status/381021802941906944

Comments

- Code should explain itself
- Avoid noisy code
  - You have source control for "metadata" or old code
  - Comments should add value

```
// Find item price
procedure FindItemPrice(ItemNo: Code[20]): Decimal
```



Comments

- XML Comments

  Additional help if you are creating libraries

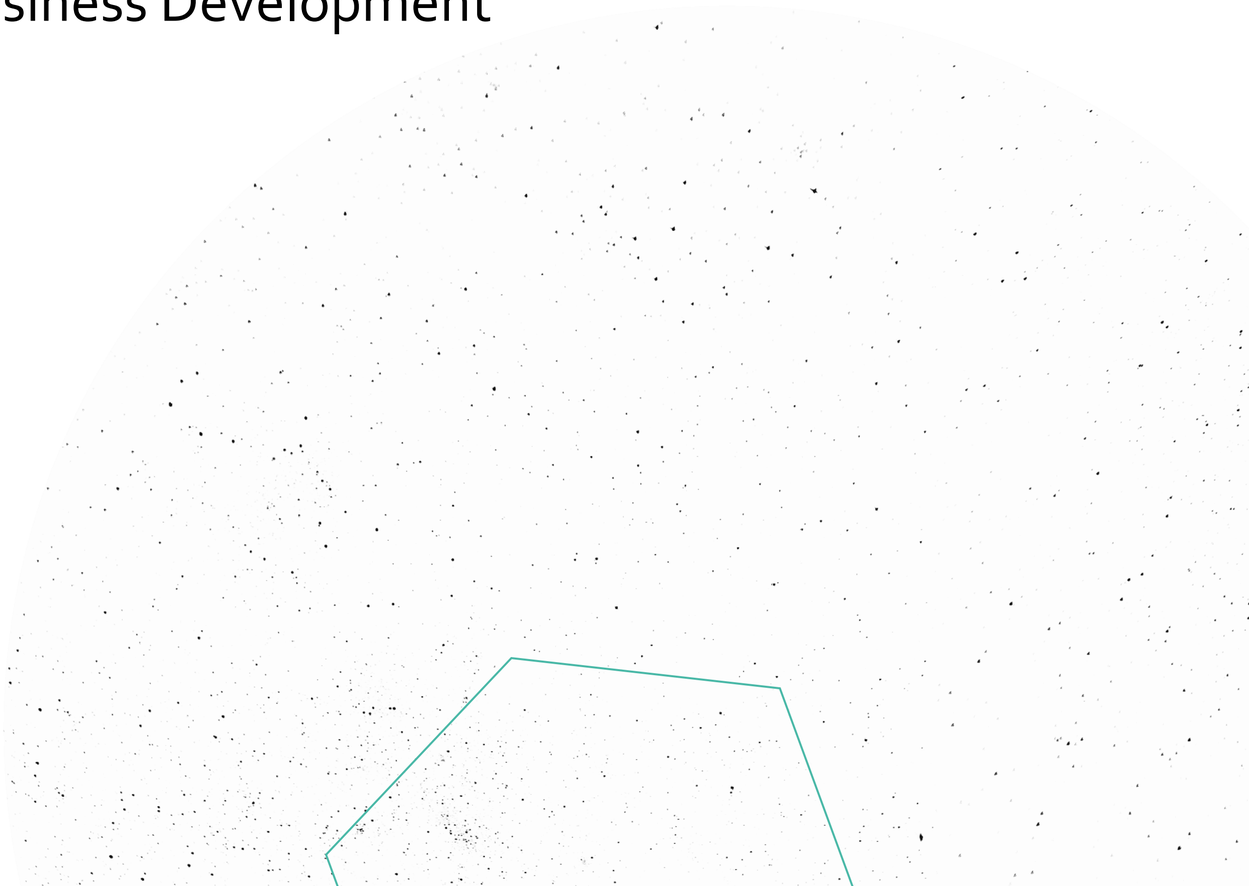
  “AL Xml Documentation” extension by “365 Business Development”

- TO-DO Comments

  Todo Tree VS Code Extension

- Mandatory comment next to COMMIT

  BusinessCentral.LinterCop (Stefan Maron)



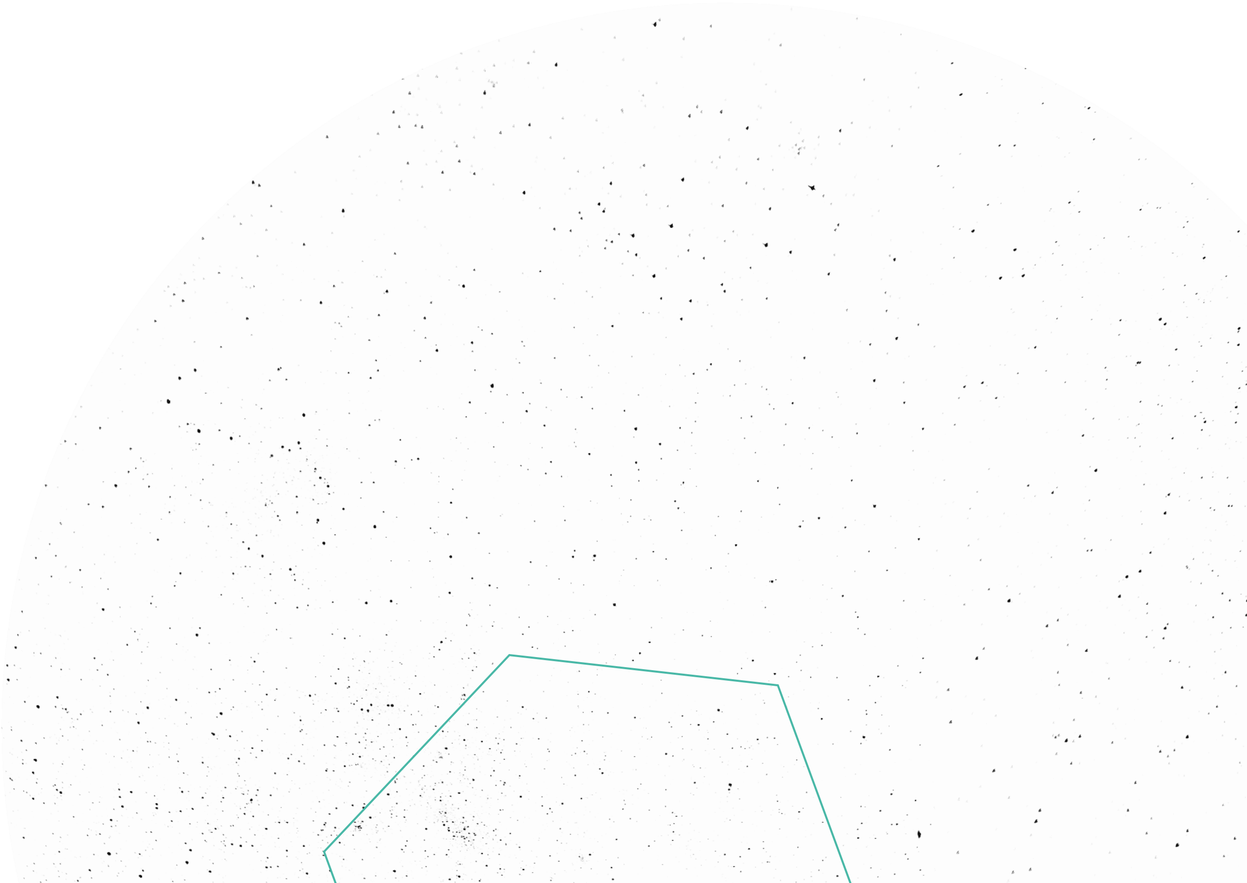
BC/NAV TECH DAYS 2022
mibuso.com

Demo

Comments

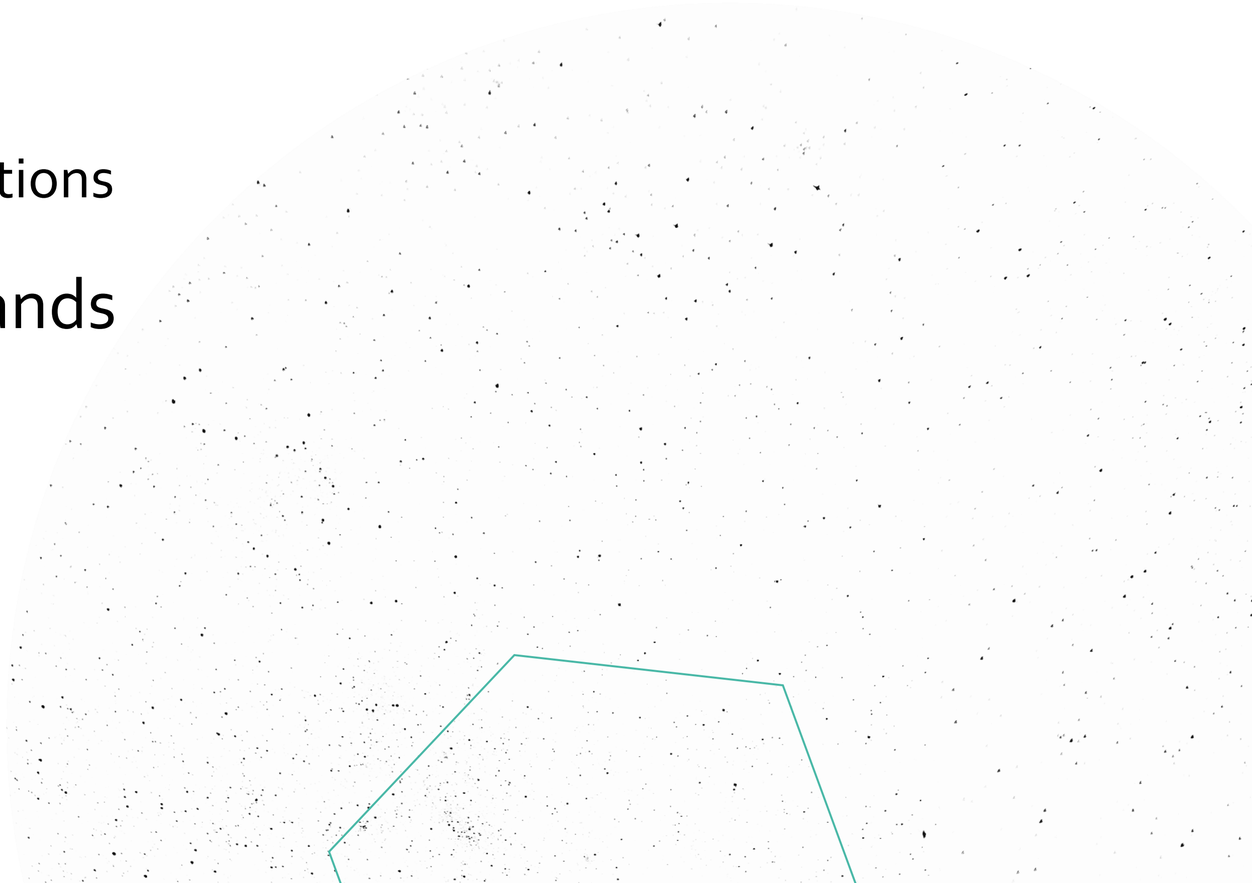
Formatting

- Our code is a document for other developers
- It could be hart to read
  - Too much noise
  - Bad formatting
  - Random order of methods



Formatting

- Variables, procedure or properties sorting
  - Does it help you?
  - Think about regions
  - Use CodeCop for variable sorting issues
  - Use AZ AL Dev Tools commands or OnSave actions
- Clean it with “AZ AL Dev Tools” commands
  - Remove empty lines or sections
  - Remove empty triggers and event subscribers
  - Fix identifiers and keywords case



BC/NAV TECH DAYS 2022
mibuso.com

Demo

Sorting
Removing empty elements

Code Analyzers

BC/NAV TECH DAYS 2022
mibuso.com

- From Microsoft
- From Community - BusinessCentral.LinterCop by Stefan Maron
  https://github.com/StefanMaron/BusinessCentral.LinterCop
- Create your own
  https://anzwdev.wordpress.com/2019/11/09/custom-al-code-analyzers/
- Don't use them if you don't want to fix warnings
- "Show Code Analyzers Rules" command from "AZ AL Dev Tools"

Change how rules are reported

- suppressWarnings property in app.json

  There is no “suppressWarningReasons”, don’t use it

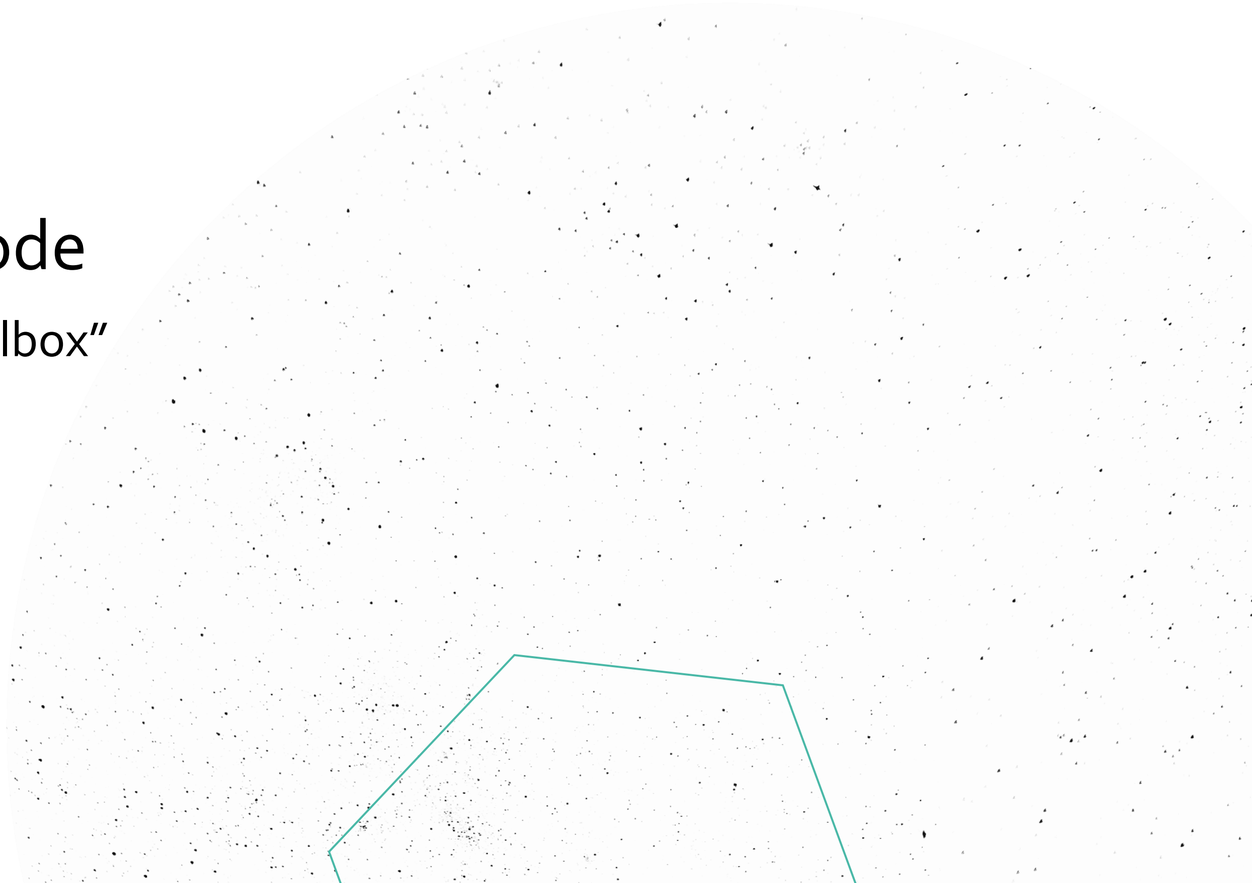
- Ruleset files

  Visual editor in “AZ AL Dev Tools”

- #pragma warning disable/restore in code
  - “Surround with Pragma” Code Action from “AL Toolbox”
  - “AZ AL Dev Tools” improvements
    - “Show warning directives” command
    - Hover and “Find all references” on rule id

- Avoid overriding safeties

  AL0432 - "{0} '{1}' is marked for removal. {2}“



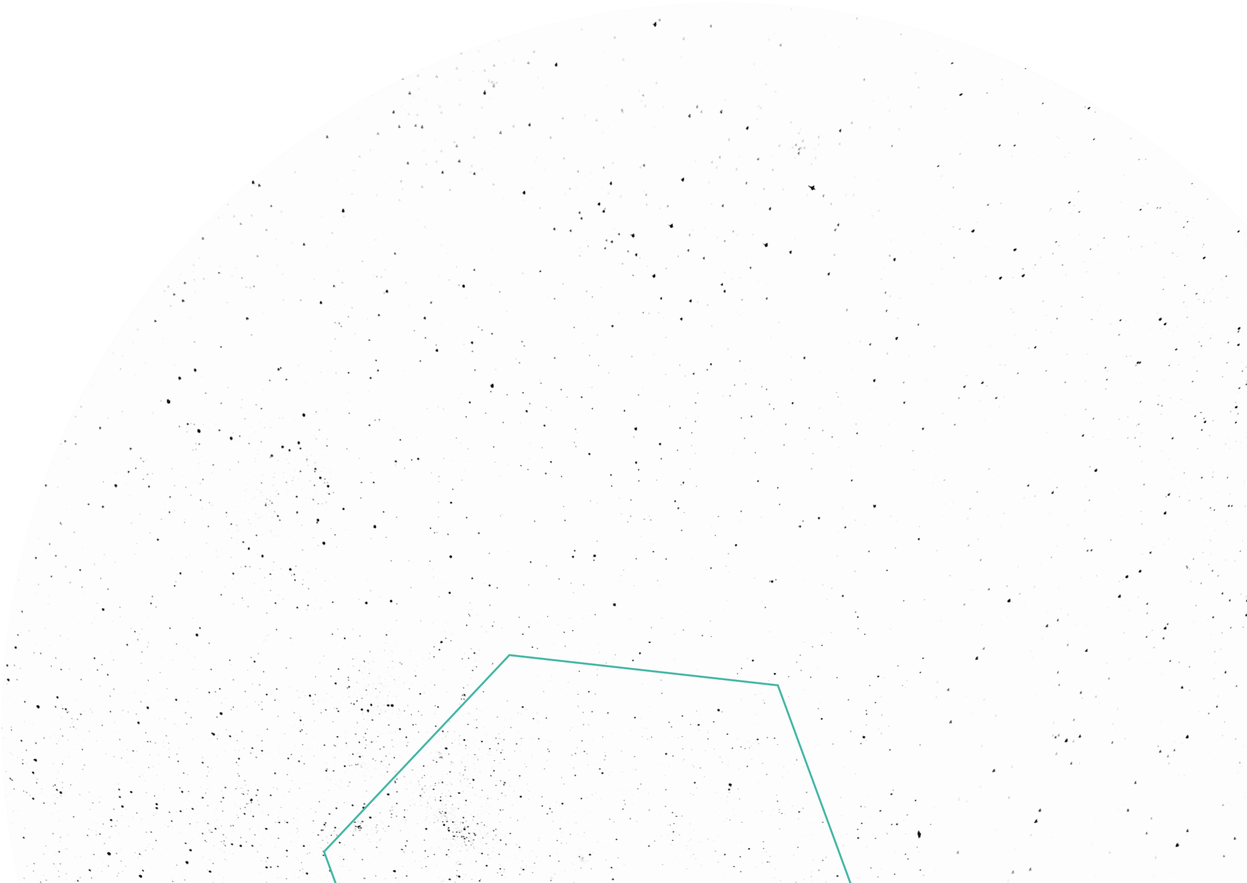
BC/NAV TECH DAYS 2022
mibuso.com

Demo

Code Analyzers

Transforming code

- Code actions
- Commands (AZ AL Dev Tools)
- Code clean-up (AZ AL Dev Tools)



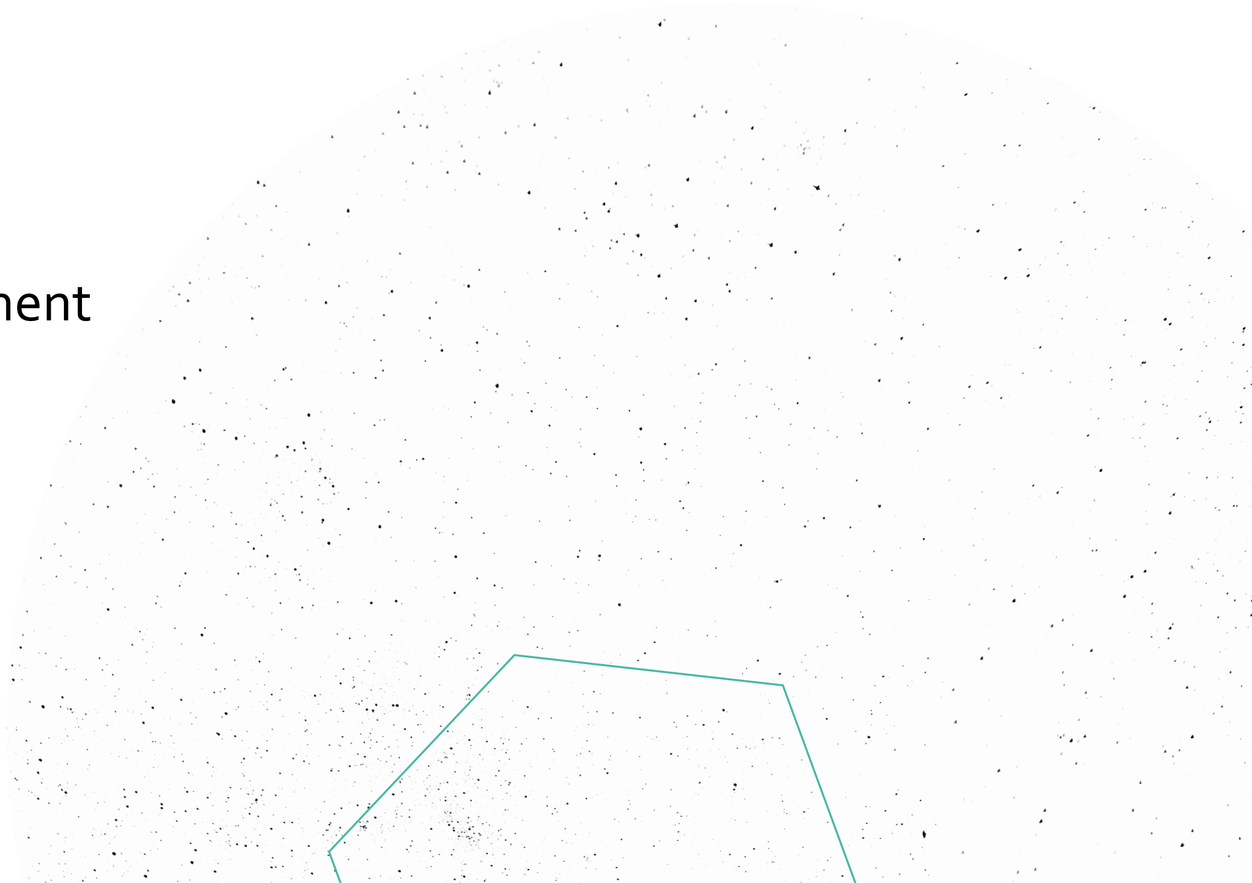
Demo

Code Cleanup

Extensions used during this session

BC/NAV TECH DAYS 2022
mibuso.com

- AL CodeActions by David Feldhoff
- AL Navigator by Waldemar Brakowski
- AL Object Ninja by Vjeko
- AL Toolbox by Bart Permentier
- AL Variable Helper by Rasmus Aaen
- AL Xml Documentation by 365 Business Development
- AZ AL Dev Tools by Andrzej Zwierzchowski
- BusinessCentral.LinterCop by Stefan Maron
- Todo Tree by Gruntfuggly
- Waldo's CRS AL Language Extension by Waldo



Any Questions?

BC/NAV TECH DAYS 2022
mibuso.com

Thank You!