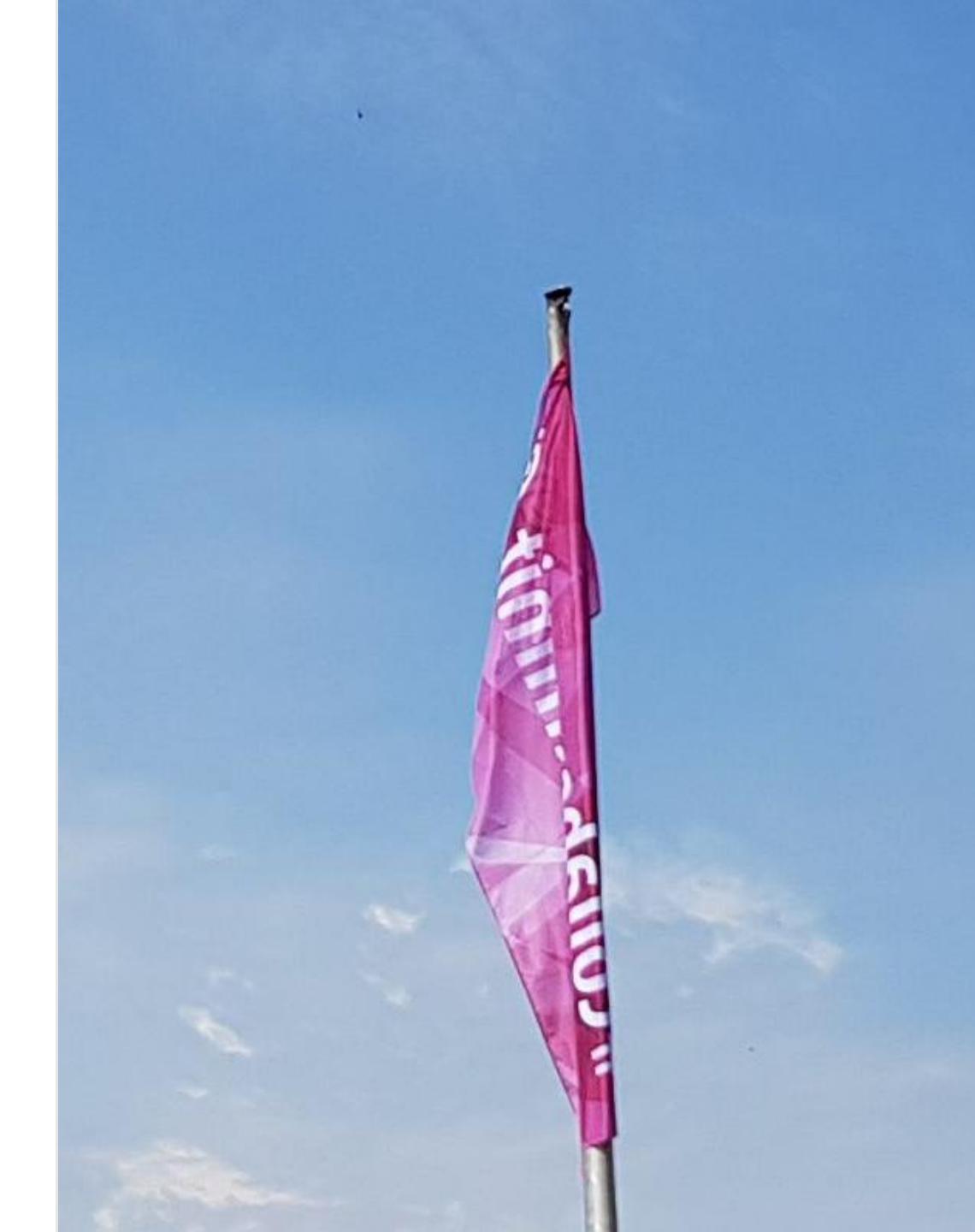




ECS 2025

✨ OPTIMIZING CANVAS APPS
FOR MAINTAINABILITY,
REUSABILITY, AND PERFORMANCE

CATHRINE BRUVOLD & DANIEL LASKEWITZ



Premium Partner



Premium Sponsor



Technology Partner



Diamond Sponsor



ShareGate:

Platinum Sponsor



Gold Sponsor



Silver Sponsor



Bronze Sponsor



Startup Silver

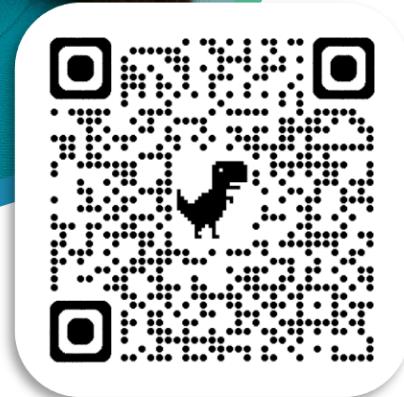


Startup Bronze



Startup Sponsor





Cathrine Bruvold

Head of Power Platform  **Point Taken**



Oslo, Norway



github.com/cathrinebruvold



linkedin.com/in/bruvold



Daniel Laskewitz

Senior Cloud Advocate  Microsoft



Utrecht, The Netherlands



github.com/laskewitz



linkedin.com/in/laskewitz



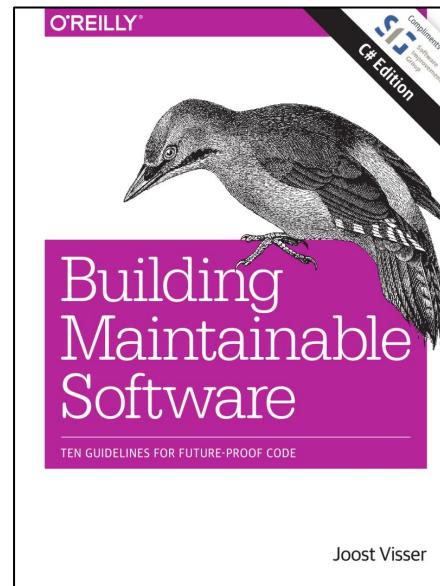
AGENDA

- ✨ Introduction
- ✨ Building Maintainable Software book
 - ✓ 10 guidelines
- ✨ Working with data
- ✨ Better performance
- ✨ Better maintainability
- ✨ Resources



Building Maintainable Software

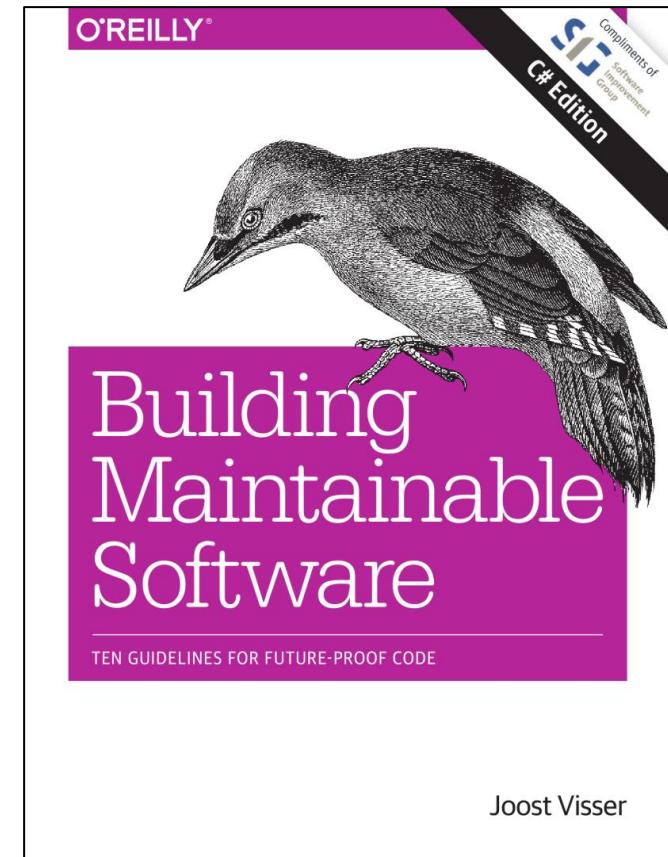
- Code principles and guidelines book -





BUILDING MAINTAINABLE SOFTWARE GUIDLINES

1. Write short units of code
2. Write simple units of code
3. Write code once
4. Keep unit interfaces small
5. Separate concerns in modules
6. Couple architecture components loosely
7. Keep architecture components balanced
8. Keep your codebase small
9. Automate development pipeline and tests
10. Write clean code



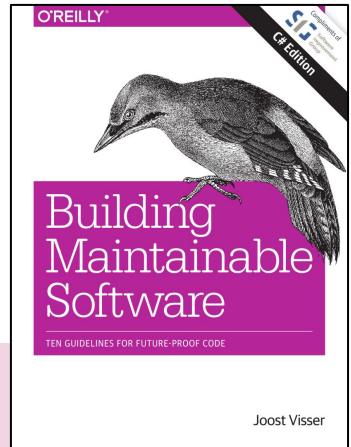


1 WRITE SHORT UNITS OF CODE



Guideline 1

- **Limit the length of code units to 15 lines of code**
- Do this by **not writing units that are longer than 15 lines of code** in the first place, or by **splitting long units into multiple smaller units** until each unit has at most 15 lines of code.
- This improves maintainability because **small units are easy to understand, easy to test, and easy to reuse.**



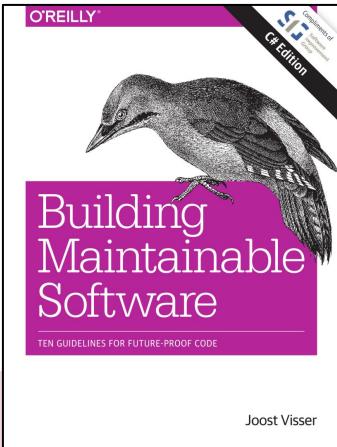


2 WRITE SIMPLE UNITS OF CODE



Guideline 2

- **Limit the number of branch points per unit to 4.**
- Do this by **splitting complex units into simpler ones** and avoiding complex units altogether.
- This improves maintainability because keeping the number of branch points low **makes units easier to modify and test**.



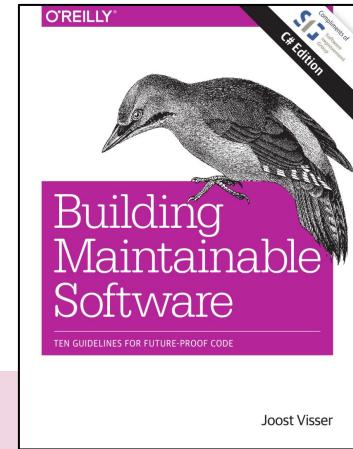


3 WRITE CODE ONCE



Guideline 3

- **Do not copy code.**
- Do this by **writing reusable, generic code and/or calling existing methods instead.**
- This improves maintainability because **when code is copied, bugs need to be fixed at multiple places**, which is inefficient and error-prone.





Working with data

- Avoid letting data take the scenic route through the internet -

① Write short units of code



POWER CAT TOOLS

DEMO

- Code review
- AI solution documentation
- Risk assessment

...and more

The screenshot shows the Power CAT Tools website interface. At the top, there's a navigation bar with 'Power Apps' and 'Power CAT Tools'. Below the navigation is a search bar. The main content area has a title 'Power CAT Tools' and a subtitle 'Accelerate solutions with tailored tools inspired by real-world development innovation.' It features several cards under the 'Features' section:

- Code review:** Improve development workstreams by automated code review with comprehensive guidance.
- AI Documentation:** Automatically generate documentation from your solutions using generative AI.
- Risk assessment:** Comprehensive risk profiles and mitigation strategies based on predefined customizable baseline risk ratings.
- Plugin monitor:** Helps create Power App experiences on the web and mobile platforms.
- API playground:** Helps create Power App experiences on the web and mobile platforms.
- Power Fx functions:** Helps create Power Apps experiences on the web and mobile platforms.

Below the features are sections for 'Featured kits' (Copilot Studio kit, Creator kit) and 'Learning resources' (Documentation, Frequently asked question, Support).

The screenshot shows a Microsoft blog post titled 'Elevate your Microsoft Power Platform development experience with Power CAT Tools' by Denise Moran, published on March 4, 2025. The post has a 4-minute read time. To the right of the text is a photograph of a person sitting at a desk, working on a laptop. The background of the post features a blue and white abstract shape.



DEMO

THE APP OF HORRORS 😱

Lack of maintainability

Hard coded values, repetition of formulas, poor naming conventions

Bad performance

Unnecessary loading of data, hitting delegation limitations, nested lookups and galleries

Little to no reusability

Code duplication, repetition and spaghetti logic

User experience

Confusing navigation, information overload and static layouts

IO-nummer	Datasource	VendorID	VendorName	Modified	MaPros	FeilType	Lokasjon
1376539				13.05.2024 - 09.40.03	No	FailedNetTotalAmountValid...	355
6569846				10.05.2024 - 15.40.02	No	FailedNetTotalAmountValid...	124
1377252				13.05.2024 - 11.10.05	No	FailedNetTotalAmountValid...	302
6570552				13.05.2024 - 11.25.04	No	FailedLineDeliveryDateValid...	124
1377195				13.05.2024 - 10.05.03	No	FailedNetTotalAmountValid...	768
1377177				13.05.2024 - 10.20.04	No	FailedNetTotalAmountValid...	765
1375133				13.05.2024 - 10.25.03	No	FailedNetTotalAmountValid...	200
1376581				13.05.2024 - 10.45.07	No	FailedNetTotalAmountValid...	372
1376288				13.05.2024 - 10.45.07	No	FailedNetTotalAmountValid...	160
6566752				13.05.2024 - 13.00.03	No	FailedNetTotalAmountValid...	760
1376813				13.05.2024 - 11.05.04	No	FailedNetTotalAmountValid...	200
1376710				10.05.2024 - 14.05.02	No	FailedNetTotalAmountValid...	765
6570553				13.05.2024 - 10.05.03	No	FailedLineDeliveryDateValid...	110
6570561				13.05.2024 - 10.05.04	No	FailedLineDeliveryDateValid...	386



Ordrebekreftelser

M3 On Prem

Replace

Søk i listen ID, leverandør og IO... Søk i database etter leverandør eller IO... Vis ubehandlete ordrer fra-dato 05.11.2024 Vis et bestemt antall 25 Hent behandlet Reset

IO-nummer	Selskap	Leverandør	ID	Endret	Behandlet	Feilmelding	Lokasjon	Antall: 25
1376579			100324	10.05.2024 - 12.05.04	Nei	FailedNetTotalAmountValidation 303	>	
6570070			100122	10.05.2024 - 12.05.04	Nei	FailedLineDeliveryDateValidation 108	>	
1376541			100324	10.05.2024 - 12.05.05	Nei	FailedNetTotalAmountValidation 160	>	
1376605								
1376486								
8170371								
8170495								
1376244								
1376586								
1376519								
1376441								
8170440								
1376431								
6570038								

Ordrebekrefte: 1376579 FailedNetTotalAmountValidation / Manuelt håndtert: Nei / Oppdatert: 10.05.2024 12.05

IO nummer: 1376579 Søk

Leverandør: Leverandør ID: JobID: 50548486 Date: 10.05.2024 13:58
Innkjøper: Avdeling: 303
Ordertype: 101 Medieprofil: 3
Leveransemetode: Z02 Incoterm: DDP
Valuta: NOK Lav/høy linjestatus: 85 / 99
Ordre ID: 50548486 Slett

Informasjon
Ordrenummer: mangler_55313645
Enhet: PAK, STK
Dato: 22.05.2024

0 - 1	Leverandørs ordrenummer	AVGIFT	KAPP	MILJØ	EMBALLASJE	LINJER	TOTALSUM	Bekrefte ✓			Sett SAI dato	Skjul info	Skjul linjer	Marker OK
□ 20	55313645	FRAKT	PLUKK	TAK	PALL	3	700							
✓ LINJE SUB ARTIKKEL ANTALL ENHET PRIS P-EHT *RAB1 *RAB2 *RAB3 LINJESUM *LEVERINGSDATO LAV HØY LEV.ARTIKKEL *ORDRE INCOT... *METODE KILDE	7842776 laminatg l0352 morning oak 1 PAK 271.11 M2 26.5 0 0 318.03 22-05-2024 85 85 L0352-03364 DDP Z02 M3	6001001 transport 1 STK 700 STK 0 0 0 700 22-05-2024 99 99 FRAKT DDP Z02 M3	703 HEAD frakt 1 STK 700 STK 0 0 0 700 22-05-2024 85 85 FRAKT SAI/											



DEMO

THE APP OF HORRORS 😱

Reference to controls on other screens

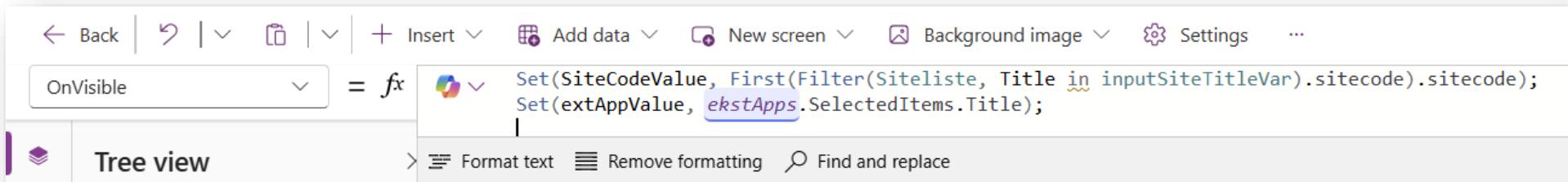
```
// Reference to controls on other screens
),
NåværendeJuridiskEier: If(eierSelskap.Value, "Company: ", "Private Individual: ") & Tlf_Eier.Text,
LeveringsadresseForSIMKort: adresseFornavn.Text & ", " & adresseEtternavn.Text & ", " & adresse.Text & ", " & Land.Text & ", " & postNummer.Text & ", " & selskap.Text
}, JSONFormat.IndentFour & JSONFormat.IncludeBinaryData & JSONFormat.IgnoreUnsupportedTypes
```

Setting too many variables

```
// Blank out any variables that were used just in case!
Set(SiteCodeValue, Blank()); Set(warehouseValue, Blank()); Set(BUvalue, Blank()); Set(BAvalue, Blank()); Set(DivisjonValue, Blank());

Set(SiteCodeValue, First(Filter(Siteliste, Title in Input_site.SelectedItems.Title).sitecode).sitecode);
Set(SiteAdressValue, First(Filter(Siteliste, Title in Input_site.SelectedItems.Title).Adresse).Adresse);
Set(SiteZipValue, First(Filter(Siteliste, Title in Input_site.SelectedItems.Title).Postnummer).Postnummer);
Set(SiteCityValue, First(Filter(Siteliste, Title in Input_site.SelectedItems.Title).Sted).Sted);
```

Complex syntax – poor readability





THINK ABOUT YOUR DATA: EFFICIENCY

- □ ×

```
First(Filter(Contacts, FirstName = "Daniel"))
```

- □ ×

```
Search(Contacts, "Daniel", FirstName)
```

- □ ×

```
LookUp(Contacts, 'First Name' = "Daniel", 'Full Name')
```



THINK ABOUT YOUR DATA: DELEGATION

Data processing

Offload tasks for data transformation to the server-side

Large datasets – 500+ records

Delegable data sources and syntax.

Delegable functions

Use Filter(), Sort(), StartsWith()

Locally processed – Collect(), ClearCollect(), Search()



Build for performance

Avoid loading unnecessary data – lazy loading



Better performance

- Reduce repetition and amount code -

Favorite **functions** and formulas



NAMED FORMULAS

- 1 Write short units of code
- 4 Keep units small

Read-only global variables

Automatic recalculation of values. No **timing-dependency** or manual variable update required.

Set on App.Formulas

Define a formula name and logic to be executed.
Reference the **short formula name** to call the logic.

powerapps

```
FormulaName = Expression;
```

Simple maintenance and reusability

Define once, **avoid redundant code** and efficient update process.



NAMED FORMULAS

```
// GET HEADER INFORMATION FROM DATAVERSE PO HEADER TABLE
varGetPOHeader :=
    ShowColumns(
        LookUp(
            'Purchase Order Headers',
            'Purchase Order No' = varPONumber
        ),
        'DSP Created On',
        'Highest line status',
        'Lowest line status',
        'Purchaser ID',
        'Media profile',
        'Order type',
        Currency,
        'Vendor ID',
        'Vendor Name',
        'Manually processed',
        'Purchase Order No',
        'Warehouse ID',
        Result,
        'DSP Last modified',
        Datasource,
        'Delivery method',
        Incoterm
    );

```

```
// get data from dataverse Purchase order header table
ClearCollect(
    colCurrentHeadCollection,
    varGetPOHeader // Set by named formulas
)
```

2 Write simple units of code

4 Keep units small



WITH()

Declarative temporary variables

Values are calculated **once** within the context of the With() block of code.

Benefits

Better readability of complex formulas

powerapps

With(

{ name1: value1, name2: value2 },

expression_using_name1_and_name2

)

2 Write simple units of code

4 Keep units small



WITH()

Old syntax

```
(RateSlider.Value / 8 / 100 / 12 * AmountSlider.Value * 10000) /  
(1 - ^ (1 + RateSlider.Value / 8 / 100 / 12, -YearsSlider.Value * 12))
```

- □ ×

Improved syntax

```
With(  
    { r: RateSlider / 8 / 100 / 12,           // rate  
      p: AmountSlider * 10000,                 // principal  
      n: YearsSlider * 12                      // terms  
    },  
    r * p / (1 - ^ (1 + r, -n))             // payment  
)
```

- □ ×

- 1** Write short units of code
- 3** Write code once
- 5** Separate concerns in modules



USER DEFINED FUNCTIONS/TYPES

Blocks of logic

Custom functions that can be called from anywhere in the canvas apps – extension of Named Formulas

Creating the formula

Define name, logic with dynamic parameters and data types, then the return value



Benefits

Ease of maintainability and reusability by utilizing a modular development approach. Create your own defined parameters

The screenshot shows the 'Updates' section of the Microsoft Power Apps Settings. It includes a search bar, a list of status filters (New, Preview, Experimental, Retired), and a toggle switch for 'Smart email address validation'. Below this, the 'User-defined functions' section is shown, which describes UDFs as named formulas with parameters for logic reuse, and has an 'On' toggle switch. The 'Faster offline synchronization' section is also visible, with its own toggle switch.

Settings

- General
- Display
- Copilot
- Updates**
- Support

Updates

New Preview **Experimental** Retired

Smart email address validation

Enables smart validation for Email fields.

Off

User-defined functions

User-defined functions (UDFs) are named formulas with parameters for logic reuse.

On

Faster offline synchronization

Allows offline synchronization to only download entity columns that are referenced in the app when using an auto-generated offline profile.

Off



USER DEFINED FUNTIONS/TYPES

✍️ Syntax

```
FunctionName( [ ParameterName1: ParameterType1 [ , ParameterName2: ParameterType2 ... ] ] ) :  
ReturnType = Formula;
```



📚 Example

```
Multiply(x: Number, y: Number): Number = x * y;
```





Better maintainability

- Modular building blocks -

2 Write simple units of code



RESPONSIVE LAYOUT MANAGEMENT

Containers
Horizontal and vertical
Dynamic resizing

Align and center
`Parent.Height & Parent.Width`
`Parent.Width/2-Self.Width/2`

The screenshot shows the Power Apps canvas interface. A central screen displays a user interface with sections for "My Sessions", "My Feedback", and "Agenda". The "Properties" pane on the right is open, showing settings for a container named "Container10". The "Display" tab is selected, showing the following properties:

- Direction: Horizontal
- Justify (horizontal): Center
- Align (vertical): Stretch
- Gap: 10
- Horizontal Overflow: Hide
- Vertical Overflow: Hide
- Wrap: Off

The "Size" section shows a width of 1297.7 and a height of 729.6. The "Color" section shows a black border. The "Properties" pane also lists other components like "Container8", "RESPONSIVE HTML/CSS", and "SELECT". The "Tree view" pane on the left shows the hierarchical structure of the app's components.

3 Write code once



CANVAS COMPONENTS

Reusable components

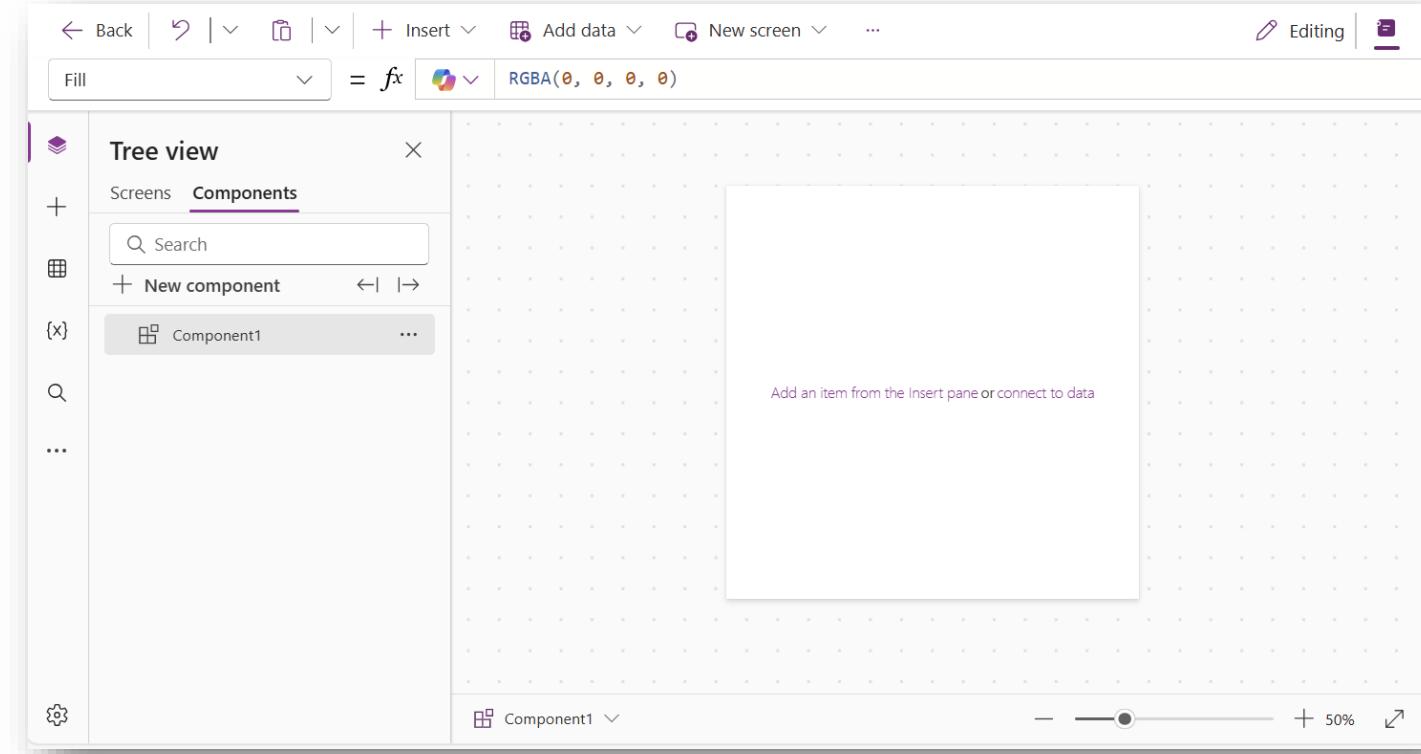
Building blocks for similar patterns.
Component library.

Implement advanced features

Custom properties – standardizing UI

Easy maintenance

Change the component once – updates
everywhere its being used... after it's published



3 Write code once



CODE COMPONENTS

Fusing code and low-code

HTML, CSS and TypeScript

Extend standard functionality

Rendered UI component that is fully customizable

Examples:

- Resco Power Components (ISV)
- Creator Kit (Microsoft Power CAT)
- PCF Gallery (Community)



Power Platform CLI

Increased complexity and can't be generated in a canvas app

A screenshot of the Resco Power Components website. The header includes links for High contrast, Customer support, Resco Power Components, Library, Resources, Pricing, and Request component. The main title is "Power Apps Components Library". Below it, a sub-instruction reads: "Browse our library of components or use the search bar to find something specific. If you don't find what you need, request a new component." A search bar with placeholder text "Enter keywords to search" is present. At the bottom, there are tabs for "Complete library" (which is highlighted in blue), Starter Components, Essential Components, and Advanced Components. Below these tabs are three preview cards showing examples of the components: "Starter Components" (a screenshot of a mobile app interface), "Advanced Components" (a screenshot of a complex dashboard or report), and "Essential Components" (a screenshot of a simple form or card).

3 Write code once

5 Separate concerns in modules



CUSTOM CONNECTORS



Reduce redundancy

Expose only what the app needs from APIs

Maintainability

APIs changes - only the custom connector needs updating.
Replaces messy HTTP requests by named actions

Security

Keep hard coded values and credentials out of apps
and flows

The screenshot shows the 'Create new custom connector' dialog in the Microsoft Power Automate interface. The top navigation bar includes 'Connector Name' (set to 'Custom Connector of Awesome'), 'General', 'Security', 'Definition' (selected), 'AI Plugin (preview)', 'Code', 'Test', and buttons for 'Swagger editor', 'Create connector', and 'Cancel'. The main area is divided into sections: 'Actions (1)', 'References (0)', 'Policies (0)', and a dropdown menu for 'New custom connector'. The 'Actions' section contains one action entry with a warning icon and a 'New action' button. The 'References' section is empty. The 'Policies' section is also empty. The 'New custom connector' dropdown lists several options: 'Create from blank', 'Create from Azure Service (Preview)', 'Import an OpenAPI file', 'Import an OpenAPI from URL', 'Import a Postman collection', and 'Import from Github'. On the right, the 'General' settings panel is visible, showing fields for 'Summary', 'Description', 'Operation ID' (marked as required), and 'Visibility' (set to 'none').



CANVAS YAML

3 Write code once

DEMO

Abstraction layer

Canvas Apps source code

YAML code

Simply and efficiently create controls and templates for apps



Benefits

Access underlying code and simplify the development process

Code

```
1 - MainContainer1_1:
8   Children:
9     - ctn_LineHeaders_3:
26       Children:
100      - lbl_ManueltProsessertHeader_3:
109        OnSelect: |+
110          =UpdateContext({ucShowProcessed: !ucShowProcessed});
111          UpdateContext({ucSortColumn:"manuallyProcessedInApp"});
112          UpdateContext({ucSortOrder:!ucSortOrder});
113
114        Size: =12
115        Text: =If(ucShowProcessed,"✓ Processed","✗ Processed")
116        Tooltip: ="Klikk her for å vise eller skjule behandlede ordrer"
117        Width: =90
118        - Icon15_3:
119          Control: Classic/Icon@2.5.0
120          Properties:
121            BorderColor: =RGBA(0, 18, 107, 1)
122            Color: =RGBA(194, 155, 39, 1)
123            Icon: =Icon.Warning
124            LayoutMinHeight: =40
125            LayoutMinWidth: =25
126            PaddingBottom: =5
127            PaddingLeft: =5
128            PaddingRight: =5
129            PaddingTop: =5
130            Width: =25
131        - lbl_resultHeader_3:
```

Copy code



DEMO

CANVAS YAML

Snippets of code

Copy and paste code intelligence in canvas apps



Reuse and contribute

Use premade Power Apps snippets and upload your own!

The screenshot shows a GitHub repository page for 'powerplatform-snippets'. The repository is public and has been forked from 'pnp/powerplatform-snippets'. It contains 2 branches and 0 tags. The 'Code' tab is selected. A commit by 'cathrinebruvold' is shown, along with several other commits from different users. The repository description states: 'Repository with snippets that can easily be used in some of the Power Platform products'. The sidebar includes links for Readme, MIT license, Activity, 0 stars, 0 watching, and 0 forks. There is also a 'Releases' section indicating 'No releases published'.

cathrinebruvold / powerplatform-snippets

Code Pull requests Actions Projects Security Insights Settings

powerplatform-snippets Public forked from pnp/powerplatform-snippets

animated-svg-loading 2 Branches 0 Tags Go to file + Code About

cathrinebruvold Completed the instructions and YAML file for the snippet 3be7aeb · last week 285 Commits

.github/workflows Updated workflow last month

assets Added Power Apps icon 11 months ago

copilot-studio Update citations.yaml last month

power-apps Completed the instructions and YAML file for the snippet last week

power-automate-desktop Updated stats. 2 months ago

Readme MIT license Activity 0 stars 0 watching 0 forks

No releases published

- 9 Automate development pipeline and tests



TEST ENGINE



Power Apps Test Engine – code first

Automation testing for UI for canvas apps



Integration

CI/DC - Azure DevOps and GitHub Action



Write tests – Power Platform CLI

TypeScript



Benefits

Faster feedback loop – no manual testing needed

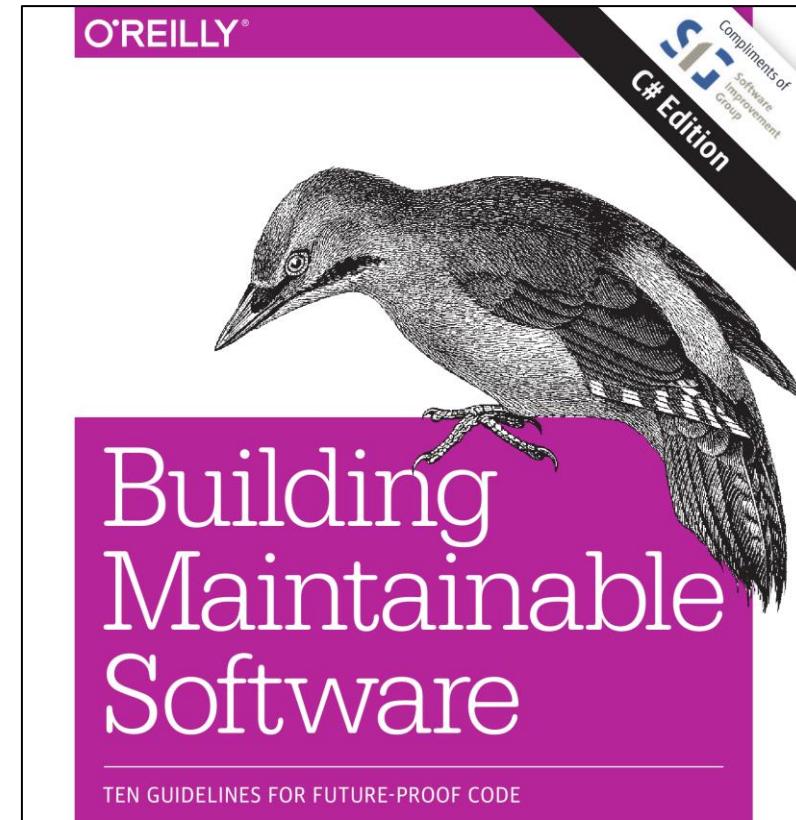
```
import test = '@microsoft/powerapps-test';

test('Submit button triggers success' as {
  async asyn.open()
  app.selectScreen("MainScreen");
  app.enterText("TextInput_Name", Cathrine);
  app.click("Button_Submit");
  app.expectText("Label_Result",Submission
successful');
})
```



BUILDING MAINTAINABLE SOFTWARE

1. Write short units of code
2. Write simple units of code
3. Write code once
4. Keep unit interfaces small
5. Separate concerns in modules
6. Couple architecture components loosely
7. Keep architecture components balanced
8. Keep your codebase small
9. Automate development pipeline and tests
10. Write clean code

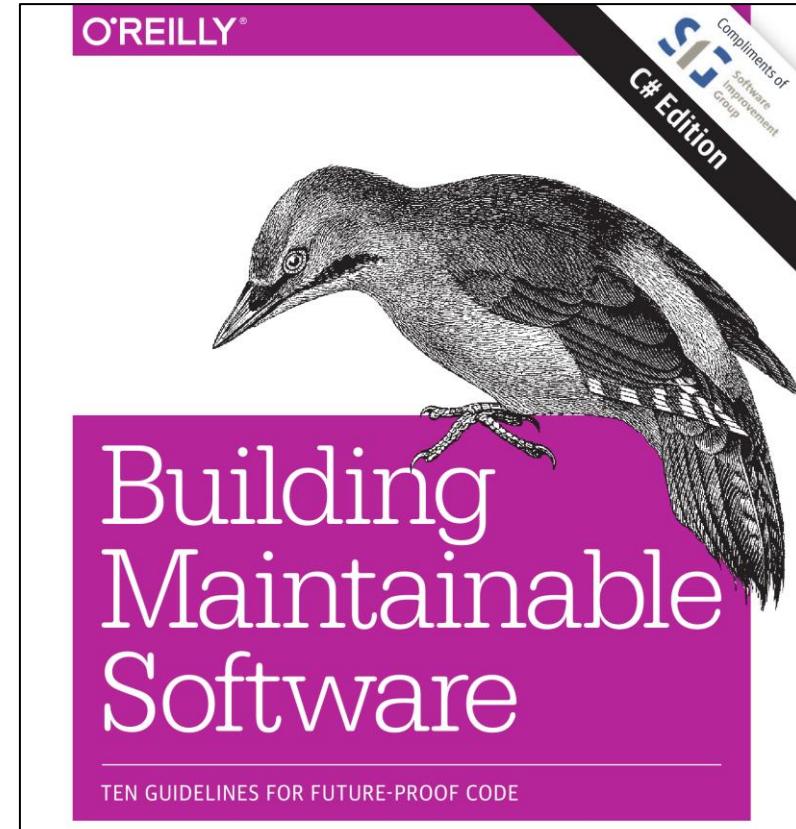


Joost Visser



BUILDING MAINTAINABLE SOFTWARE

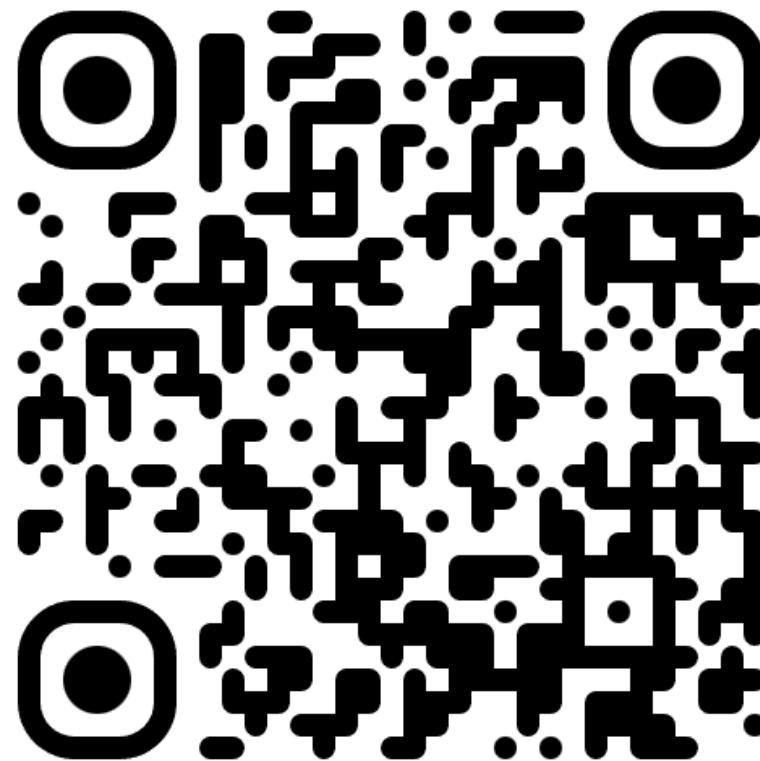
1. **Write short units of code**
2. **Write simple units of code**
3. **Write code once**
4. **Keep unit interfaces small**
5. **Separate concerns in modules**
6. Couple architecture components loosely
7. Keep architecture components balanced
8. Keep your codebase small
9. **Automate development pipeline and tests**
10. Write clean code



Joost Visser



Get our resources!





THANK YOU,
YOU ARE AWESOME ❤

PLEASE RATE THIS SESSION
IN THE MOBILE APP.

