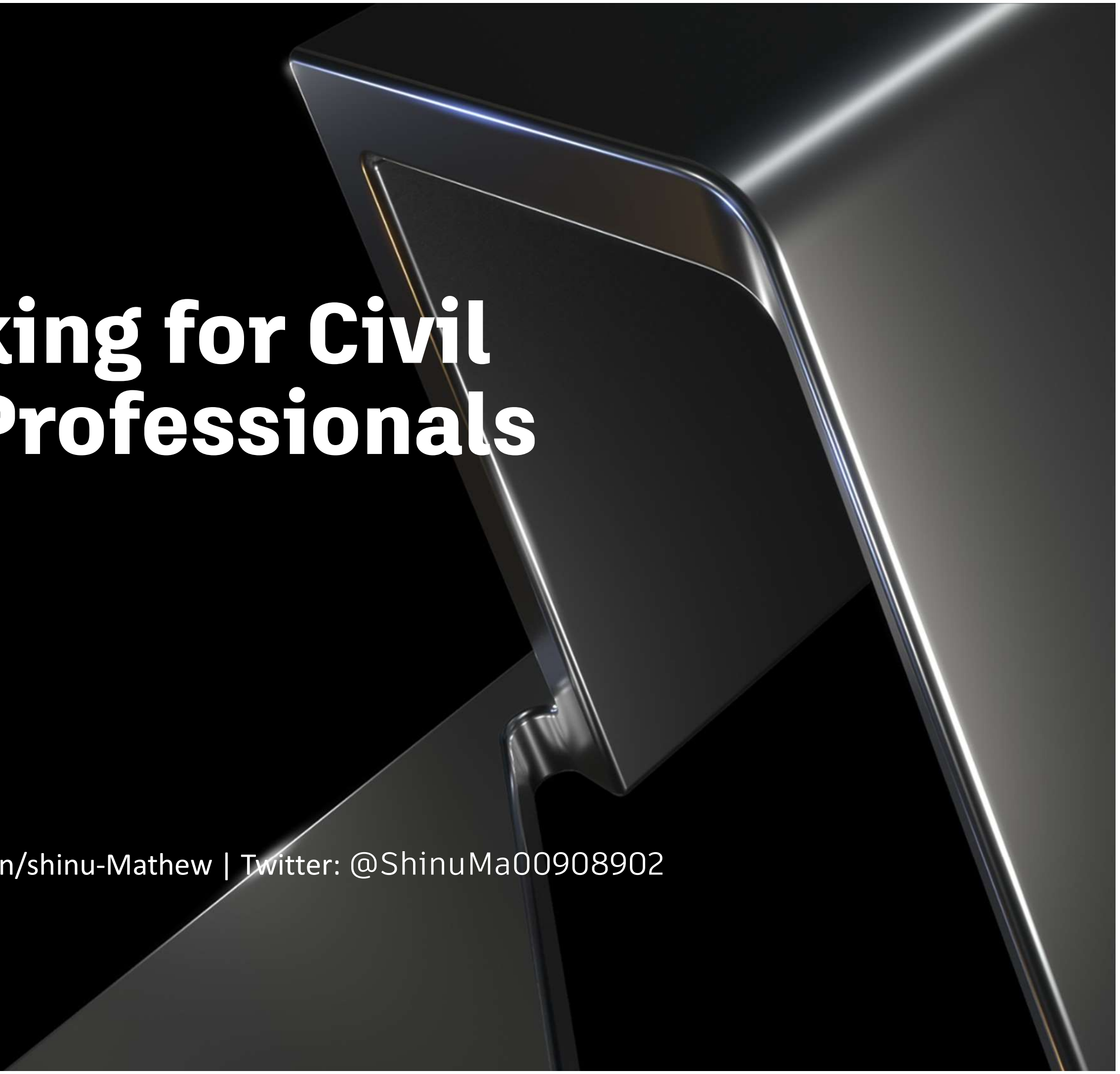
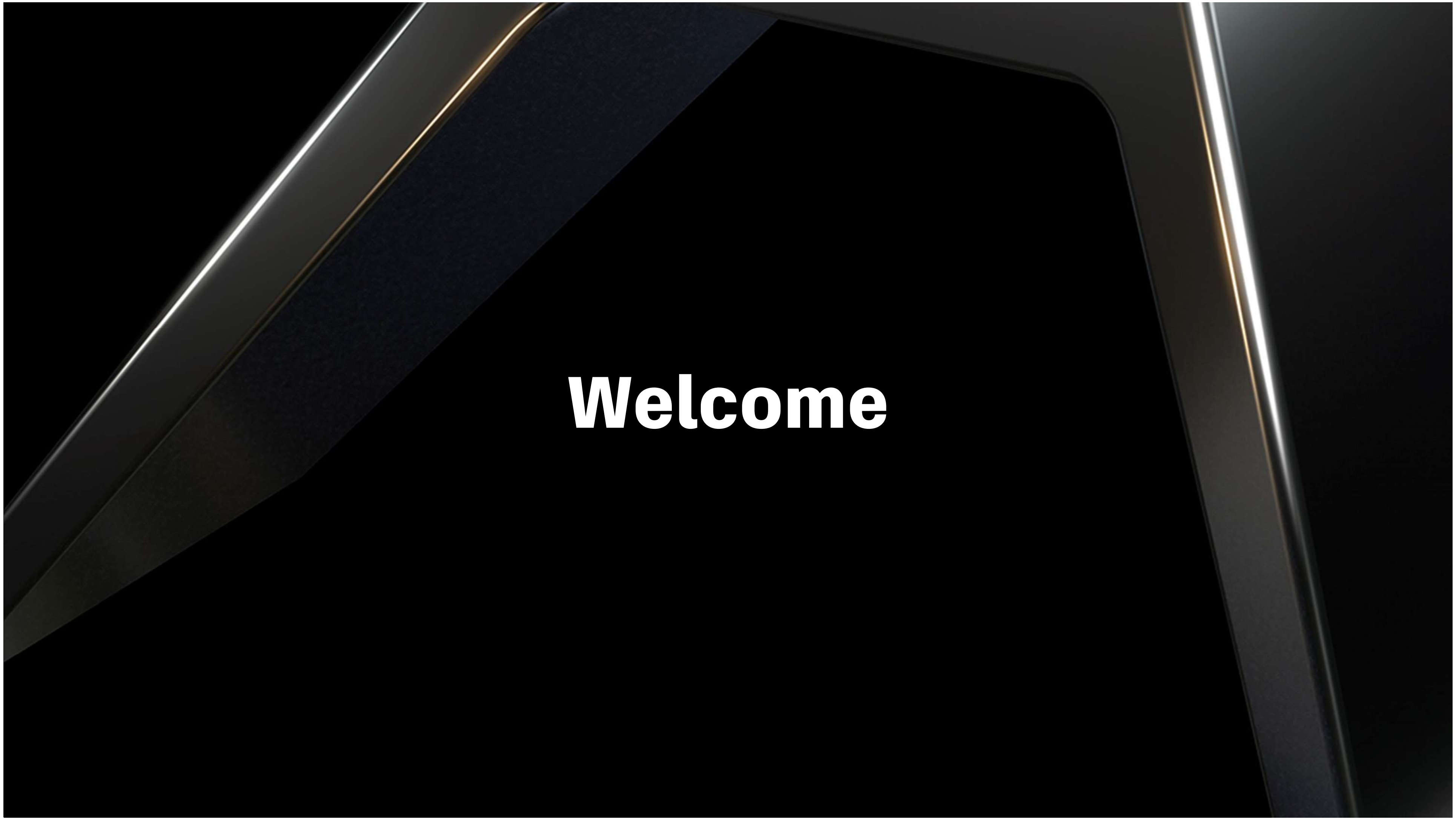
AUTODESK UNIVERSITY

# Automation Thinking for Civil Engineers & BIM Professionals



**Shinu Mathew**

Head – BIM & Allied Technologies | linkedin.com/in/shinu-Mathew | Twitter: @ShinuMa00908902



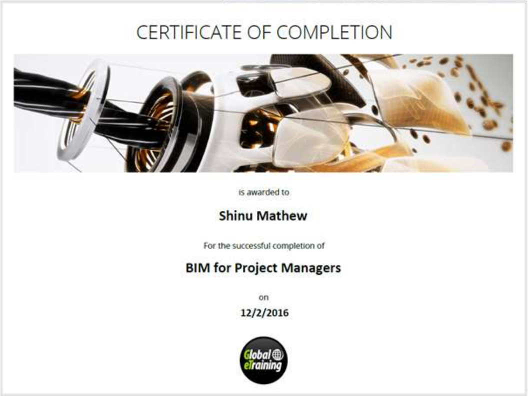
# Welcome

# About Me

Shinu Mathew
Head – BIM & Allied Technologies
Arcadis India Pvt Ltd
Bangalore

Shinu Mathew have been working in the Civil Infrastructure industry for over 24 years and of that, 9 years with Arcadis as a BIM Professional and Automation Champion.

Have developed many advanced Automations since late 1990s, using Lisp, VBA & VB.Net.



CERTIFICATE OF COMPLETION

is awarded to

**Shinu Mathew**

For the successful completion of

**BIM for Project Managers**

on

12/2/2016

# Automation Thinking - Basics

Introduction

Technology evolution in the Design & Construction industry is forcing all of us to adapt to the fast-changing digital landscape.

Why Now?

- Easier Programming Languages – Visual Programming
- Anyone Could Automate –Attitude & Aptitude
- High Demand

# Innovation

# Innovation

A presentation by Harvard Business School Professor, Clayton M Christensen, narrates the dilemma of Inventors. He talks about two types of innovations. Sustaining Innovation & Disruptive innovation.

### Sustainable

Improving the current product by;

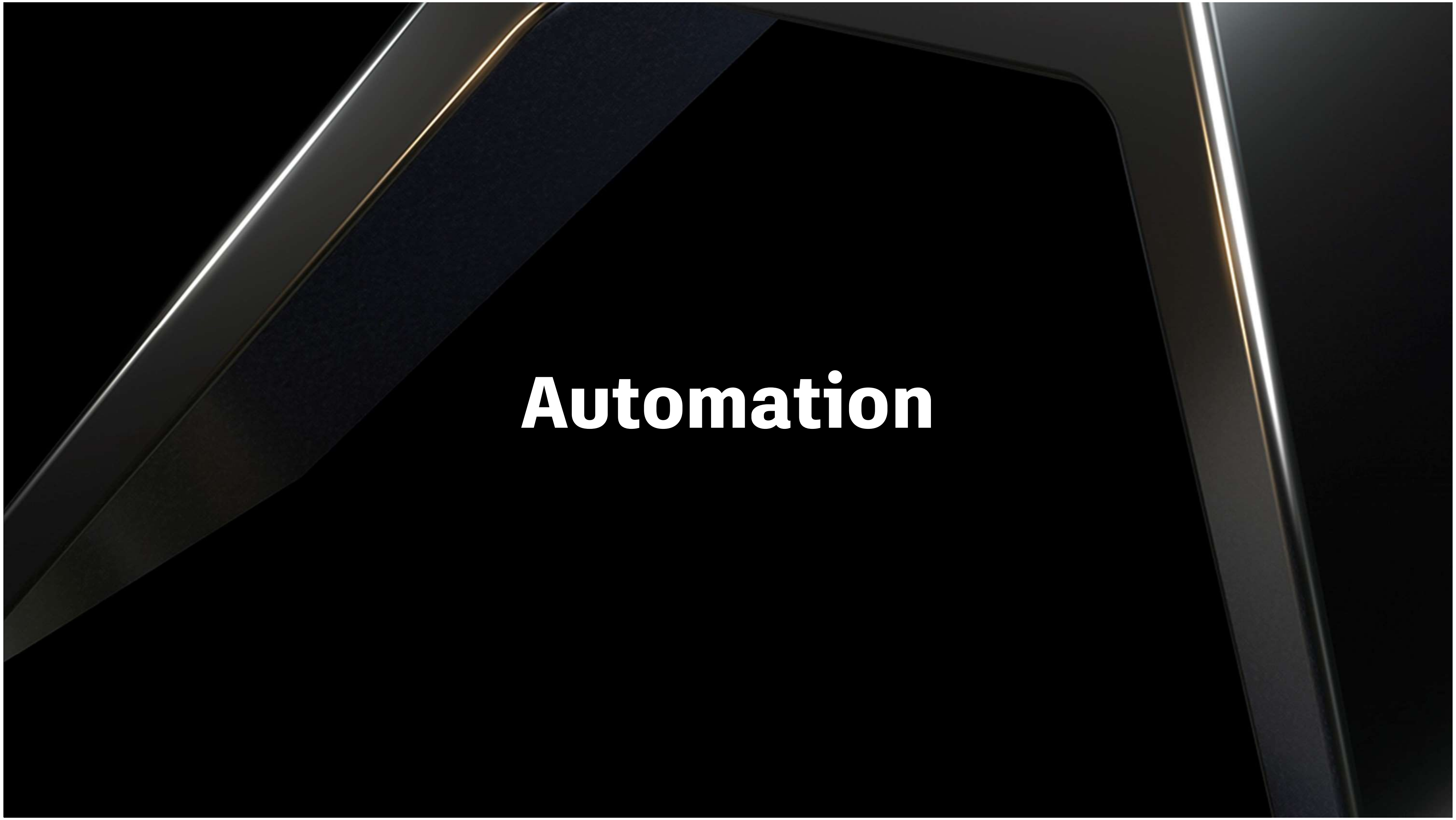
- Reducing defects
- Increasing efficiency
- Speeding up production time

Revolves around existing product portfolio & customer's current needs/expectations

### Disruptive

Often challenges the existing practices. It does not conform to the framework set by the present practices and the proponents of Sustaining Innovation would often neglect it.

- The need exists in the future
- Starts with the niche market & small clientele
- Once matured, unleashed upon the bigger market – often uprooting the market leaders



# Automation

# Standardization

We should always try to maintain a proper standard  in, but not limited to;

- Naming
- Coding
- Commenting
- Grouping
- Repository

# What is Automation?

Process with which certain tasks or processes done with minimal human intervention

Process with which certain tasks or processes done with minimal human intervention

Reducing errors & and increases efficiency

Automate **boring,** repeated tasks

# What is Automation Thinking?

Automation thinking is the process of identifying / analyzing / executing Automation opportunities

Identify

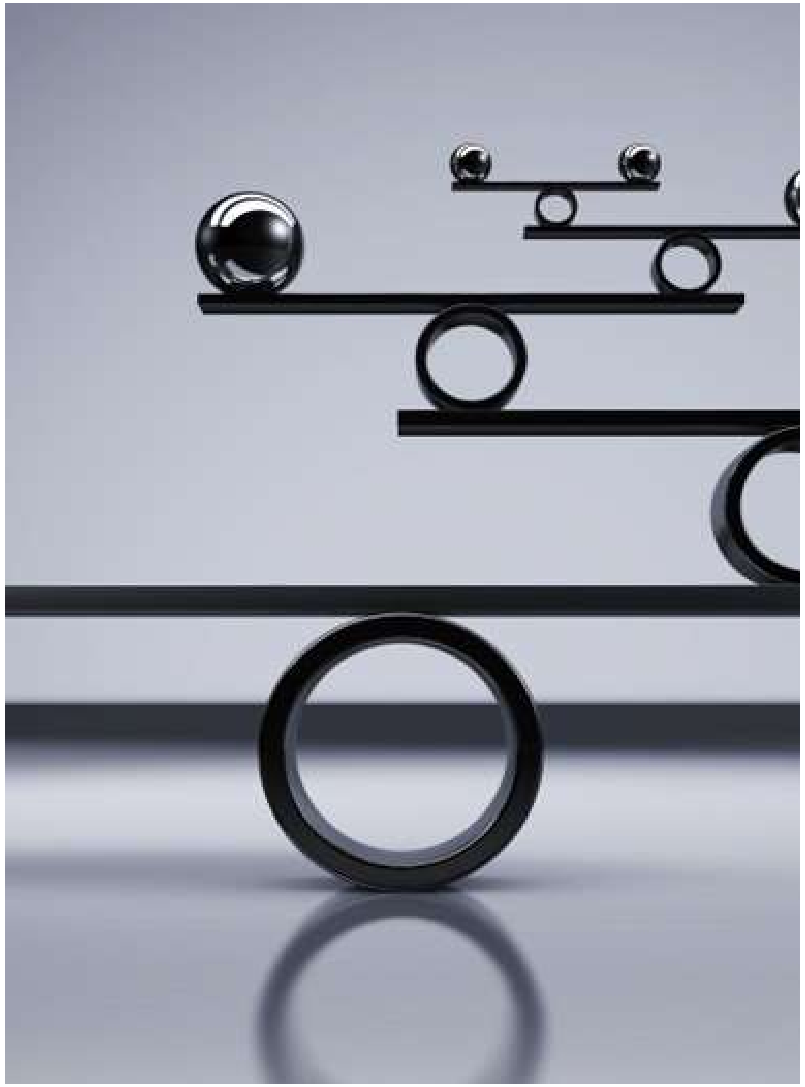
Analyze

Execute

# Background

**Evolution of design automation.**

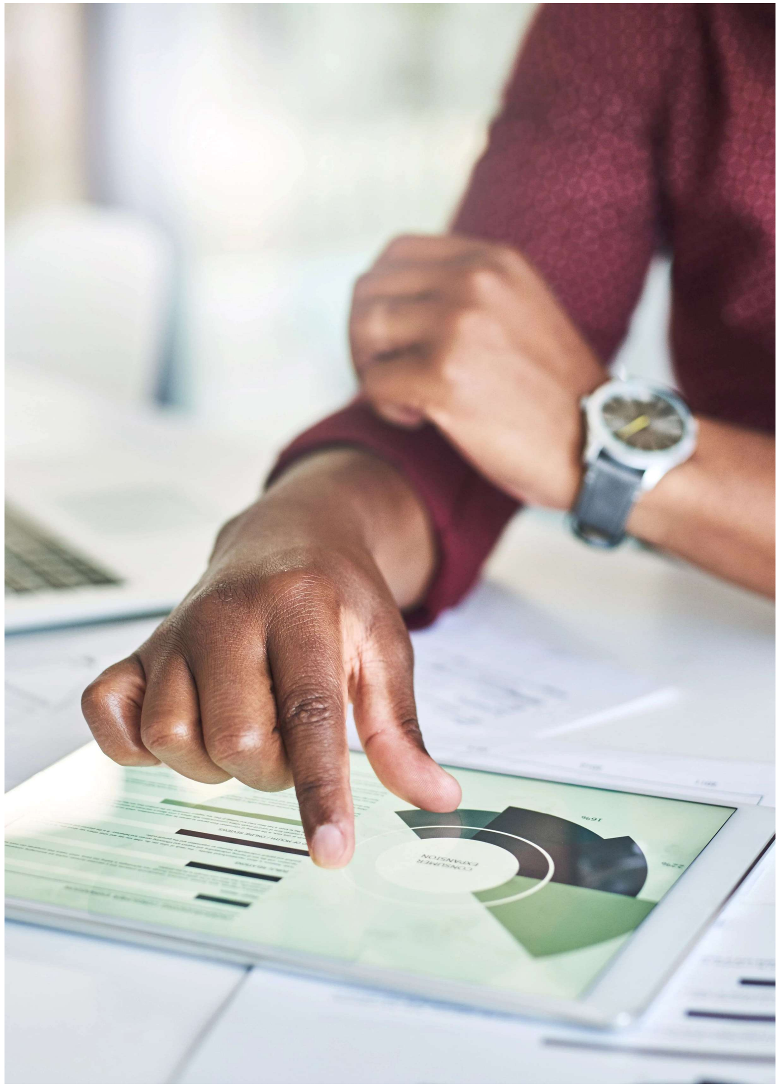
From the early days, designers found ingenious ways to automate their job. From T-Scale to Drafting machines, from plain tables to theodolites to total stations to laser scanners, they were always looking for easier & faster ways to achieve results.



# Survival of the savvy

## Automation is going to be an essential skill

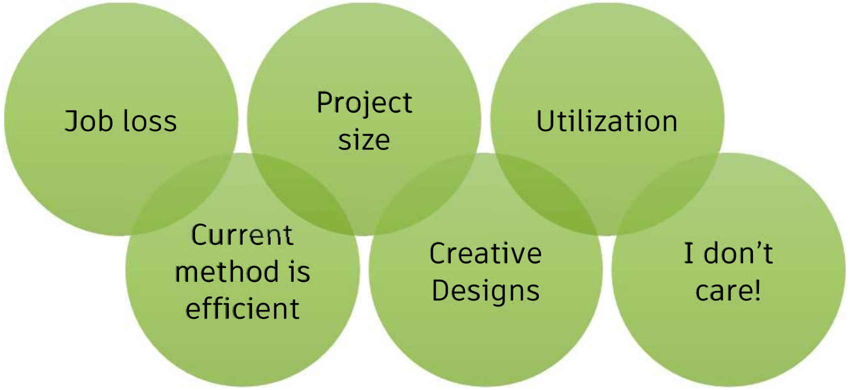
To survive and succeed, one must adapt to the changing landscape. Clinging on to the dated techniques may buy some time, but not long. Change is already here.

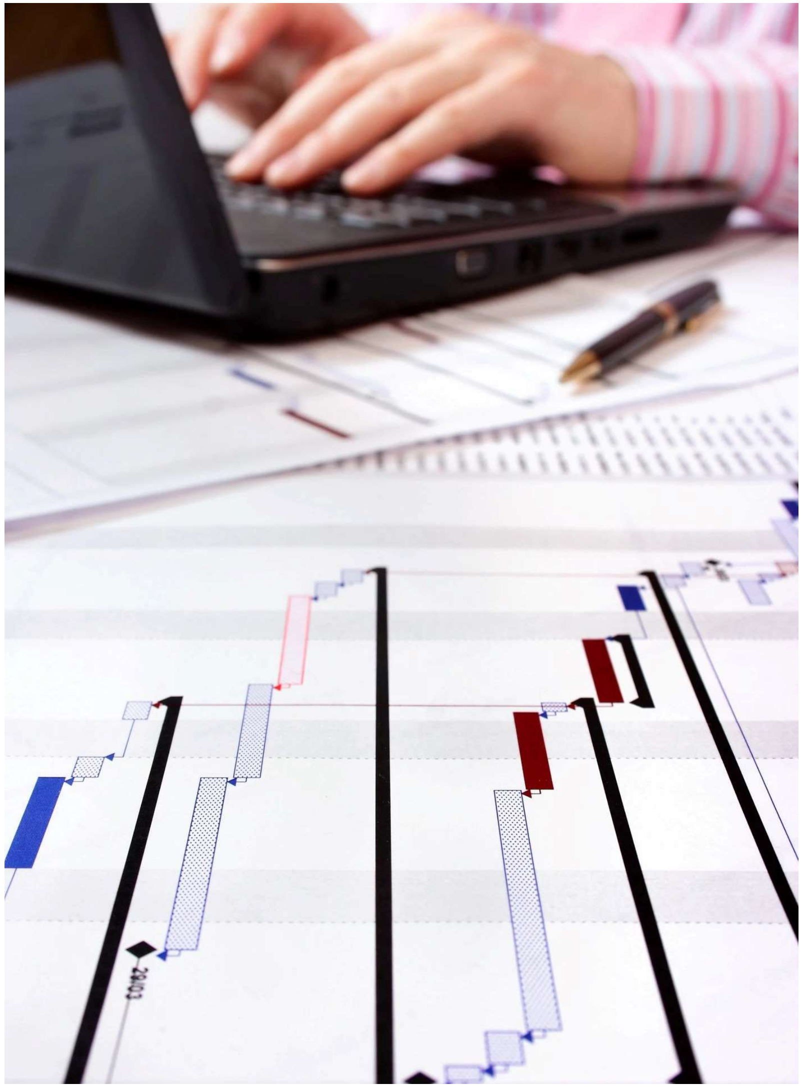


# Why Automation

**Resistance to change.**

You must have heard it many times! The pushback to change, the resistance to adopt automations among the conservative civil engineers.





# Early Examples

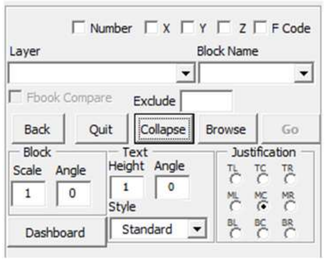
Examples from a decade ago.

### Survey points in AutoCAD

Surveyors produce CSV files, and plotting those points in AutoCAD was a tedious task. Some used Excel, some Lisp

Then I wrote a program that;

- Reads the CSV
- Determine blocks based on Feature code
- Places custom blocks & points



□ Number □ X □ Y □ Z □ F Code

Layer:

Block Name:

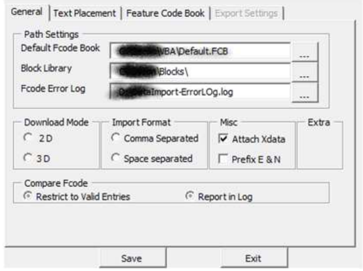
□ Fbook Compare

Exclude:

Block: Scale 1, Angle 0

Text: Height 1, Angle 0, Style Standard

Justification: ○ TL ○ TC ○ TR ○ ML ● MC ○ MR ○ BL ○ BC ○ BR



Path Settings

Default Fcode Book: [redacted]VBA\Default.FCB

Block Library: [redacted]\Blocks\

Fcode Error Log: [redacted]Import-ErrorLOg.log

Download Mode: ○ 2D ○ 3D

Import Format: ○ Comma Separated ○ Space separated

Misc: ☑ Attach Xdata □ Prefix E & N

Extra

Compare Fcode: ● Restrict to Valid Entries ● Report in Log

### Executing series of commands

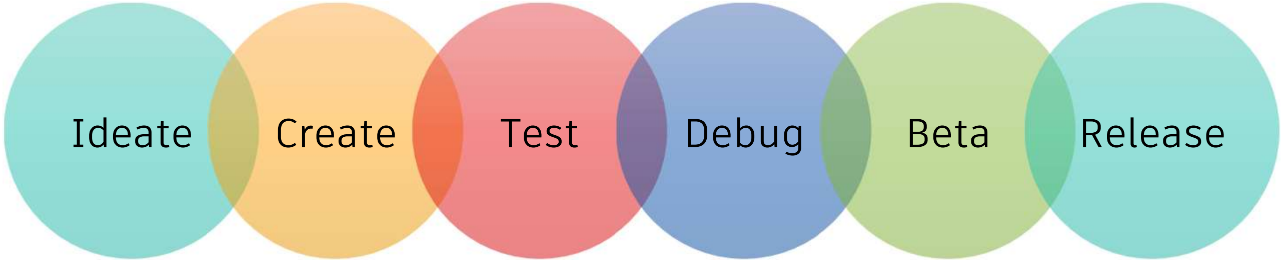
Automation by running a series of commands, as it would be if one were typing in them.

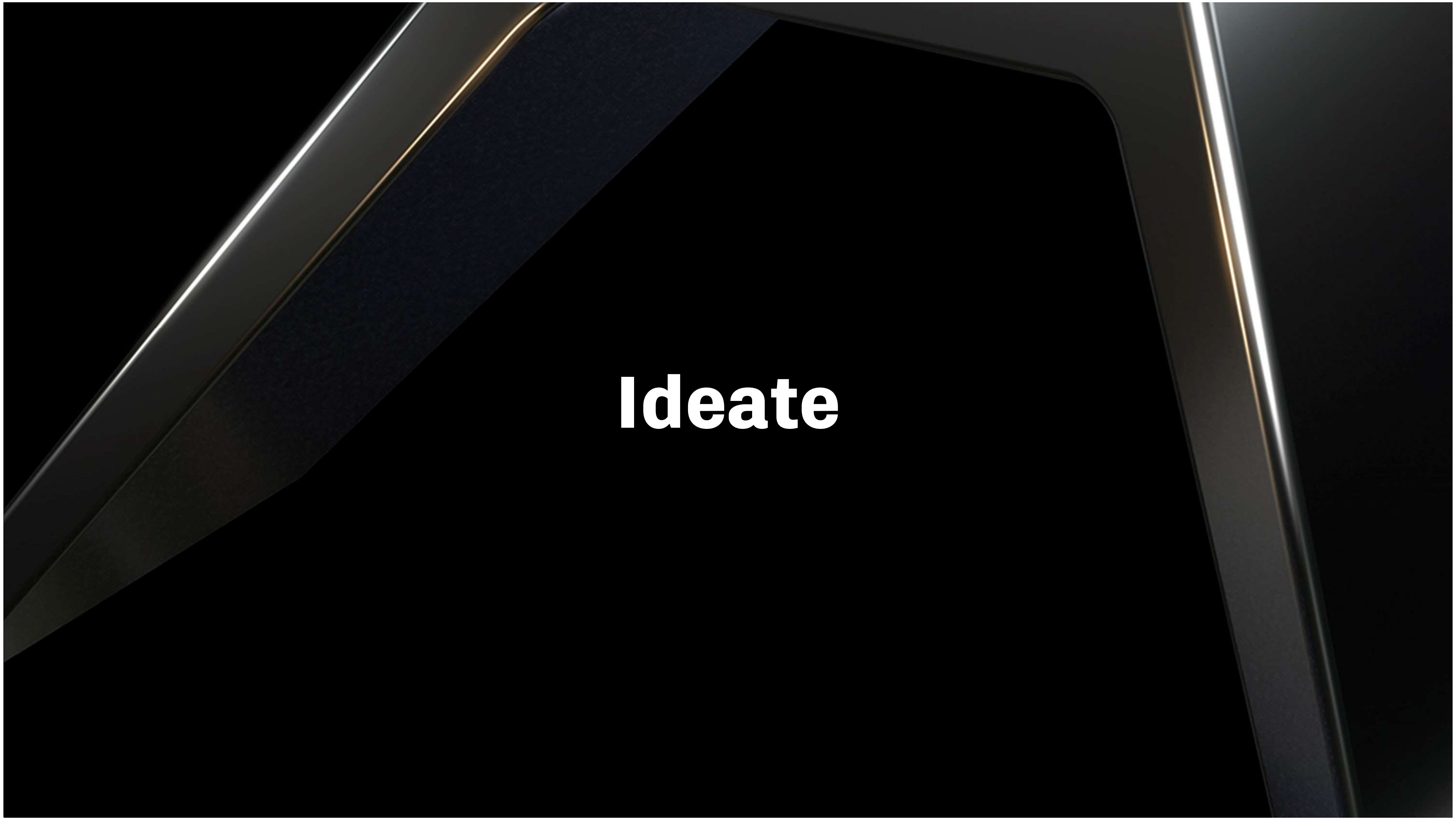
- Script files (*.scr)
- Recorded Macros

# How to Automate

# Automation – How?

To create a proper automation that can be used by a wider audience, it would require proper planning & execution. Below are some major steps in developing.

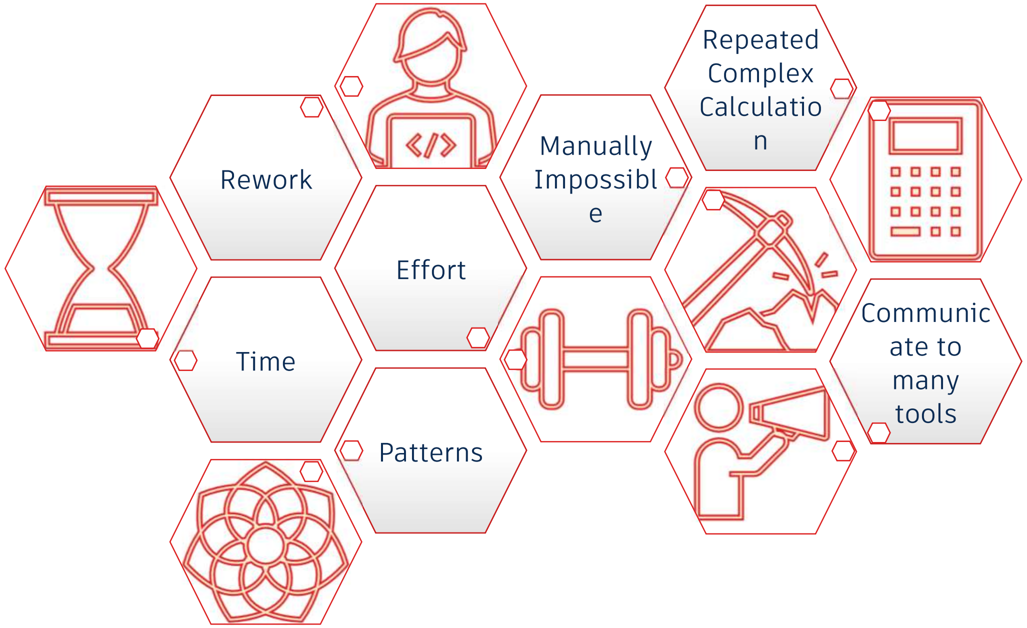




# Ideate

# Ideate

- **Questions that we should ask ourselves**



Rework

Effort

Time

Patterns

Manually Impossible

Repeated Complex Calculation

Communicate to many tools

**Find opportunity**

Identify opportunities for Automation

# Ideate

- **Who can help us?**

Automator Colleagues

Community of Practitioners

Developer Network forums

**Brainstorm/Collaboration**

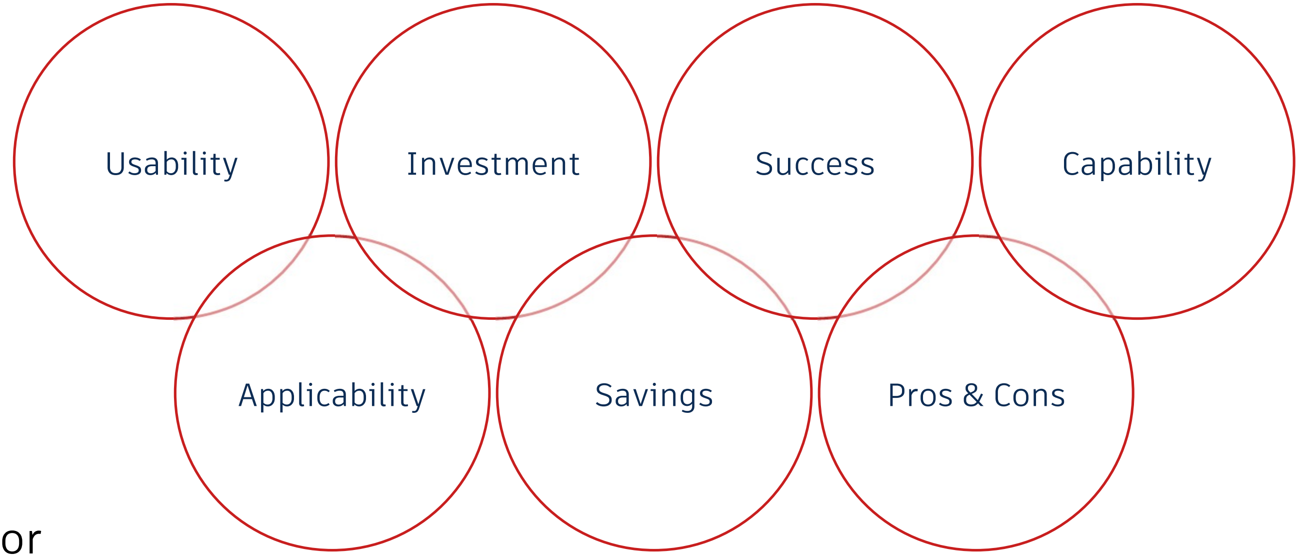
Discuss the idea

# Ideate

**Worthiness**

Identify opportunities for Automation

- **Is it worth Automating?**



Usability

Investment

Success

Capability

Applicability

Savings

Pros & Cons

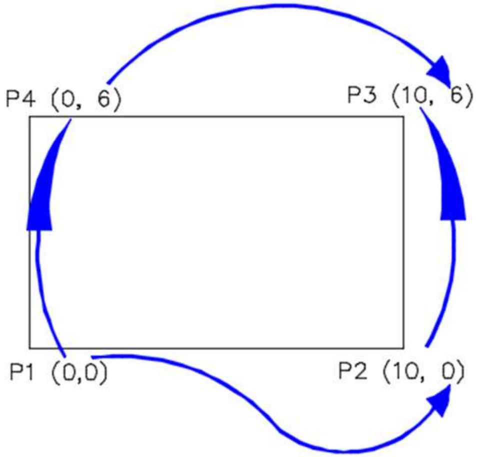
# Geometrical vs Logical

- **To create automation, embrace the logical thinking**

### Geometrical Thinking

We tend to think in Geometrical terms.

- Design Visualization is in 2D/3D
- Shapes & Sizes



### Logical Thinking

Visualizing the relation between elements.

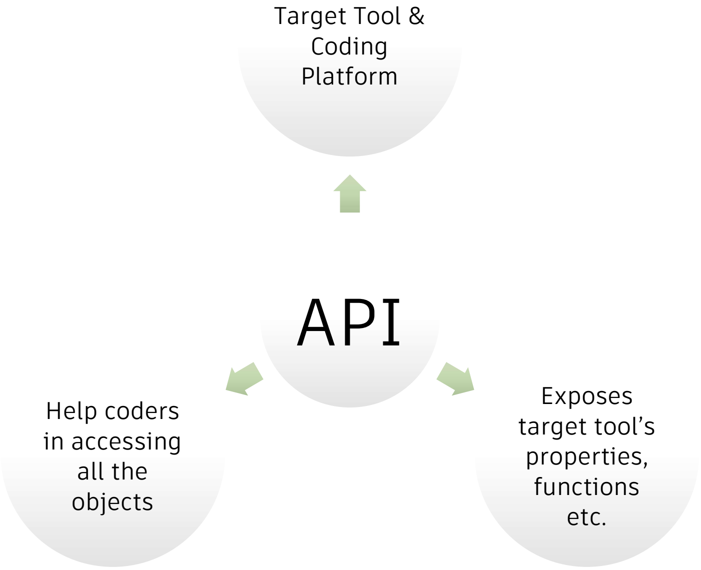
- Looking at relations & patterns
- Mathematical analysis
- Problem solving mindset

# Application Programming Interface

- **Tool-Targeted applications need API**

Application Programming interface (API) will enable the *Coding Platform* talk to the *Target Application*, by exposing the Target's internal functions to the code language.

- Connection between coding platform & targeted tool
- Exposes target tool's properties, functions etc.
- Help coders in accessing all the objects


Target Tool & Coding Platform
API
Help coders in accessing all the objects
Exposes target tool's properties, functions etc.

# Intelligence

## Branching / Conditional Logic

Intelligent decision making is important in coding. If a certain value meets a criterion, then it should execute a branch of the code, and if it does not a different one. A programmer must accommodate all scenarios for the code to cater to the need.

## Data Types

Data type is the type of a certain value / parameter with which the programmer intends to process. Each host application will have its defined types, below are the ones outside of that.

- String
- Integer
- Long
- Double / Real
- Boolean
- Datetime

## Events

A certain block of code executed when the user/application trigger an action.

- Click of a certain button on mouse / keyboard
- Click on a button/control on user interface
- Change of status of a control / object
- Addition / Deletion / change status of an element.

# End Result

## End Result

### Stand Alone

An Independent package, with no dependence on other tools

### Plugin/Add On

Hosted within another application

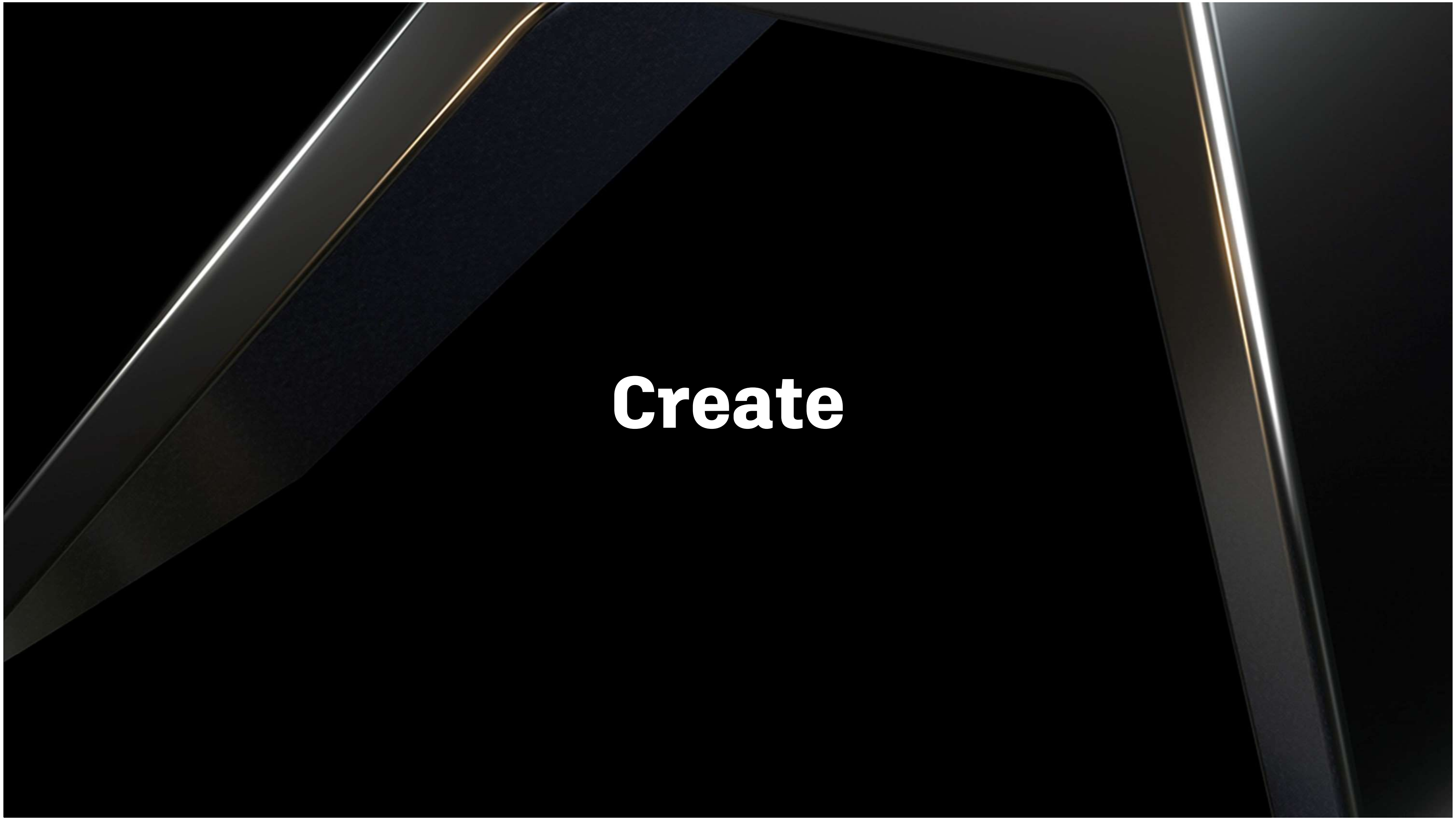
## Coding Platform

### Visual

Easy, predefined nodes

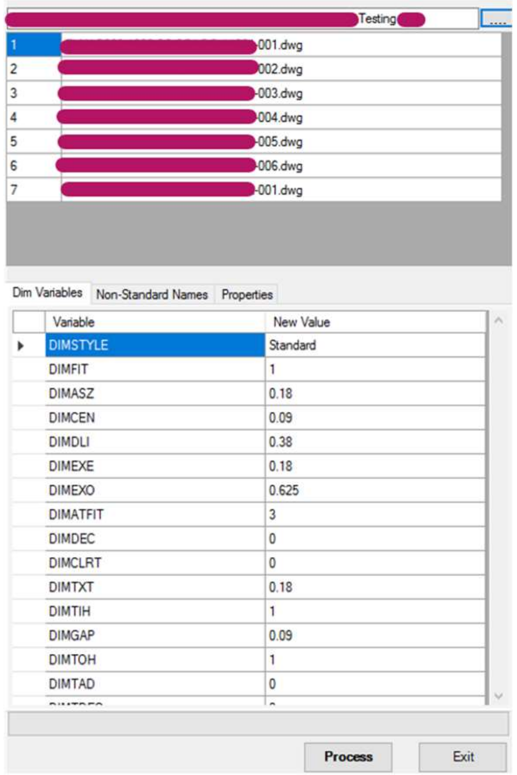
### Text

Complex programs



# Create

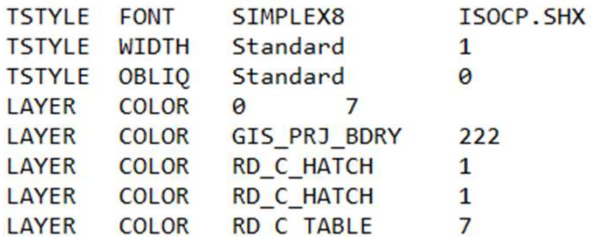
# Create



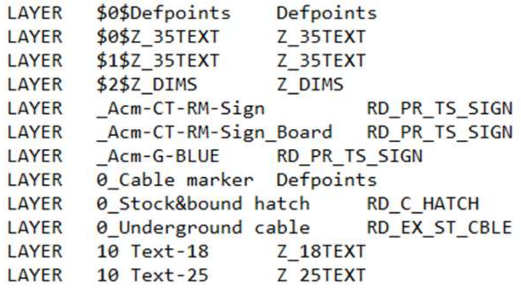
[redacted]Testing[redacted]

| | |
|---|---|
| 1 | [redacted]001.dwg |
| 2 | [redacted]002.dwg |
| 3 | [redacted]-003.dwg |
| 4 | [redacted]-004.dwg |
| 5 | [redacted]-005.dwg |
| 6 | [redacted]-006.dwg |
| 7 | [redacted]-001.dwg |

| Variable | New Value |
|---|---|
| DIMSTYLE | Standard |
| DIMFIT | 1 |
| DIMASZ | 0.18 |
| DIMCEN | 0.09 |
| DIMDLI | 0.38 |
| DIMEXE | 0.18 |
| DIMEXO | 0.625 |
| DIMATFIT | 3 |
| DIMDEC | 0 |
| DIMCLRT | 0 |
| DIMTXT | 0.18 |
| DIMTIH | 1 |
| DIMGAP | 0.09 |
| DIMTOH | 1 |
| DIMTAD | 0 |



```
TSTYLE  FONT    SIMPLEX8        ISOCP.SHX
TSTYLE  WIDTH   Standard        1
TSTYLE  OBLIQ   Standard        0
LAYER   COLOR   0       7
LAYER   COLOR   GIS_PRJ_BDRY    222
LAYER   COLOR   RD_C_HATCH      1
LAYER   COLOR   RD_C_HATCH      1
LAYER   COLOR   RD C TABLE      7
```



```
LAYER   $0$Defpoints    Defpoints
LAYER   $0$Z_35TEXT     Z_35TEXT
LAYER   $1$Z_35TEXT     Z_35TEXT
LAYER   $2$Z_DIMS       Z_DIMS
LAYER   _Acm-CT-RM-Sign         RD_PR_TS_SIGN
LAYER   _Acm-CT-RM-Sign_Board   RD_PR_TS_SIGN
LAYER   _Acm-G-BLUE     RD_PR_TS_SIGN
LAYER   0_Cable marker  Defpoints
LAYER   0_Stock&bound hatch     RD_C_HATCH
LAYER   0_Underground cable     RD_EX_ST_CBLE
LAYER   10 Text-18      Z_18TEXT
LAYER   10 Text-25      Z 25TEXT
```

**Scalability**

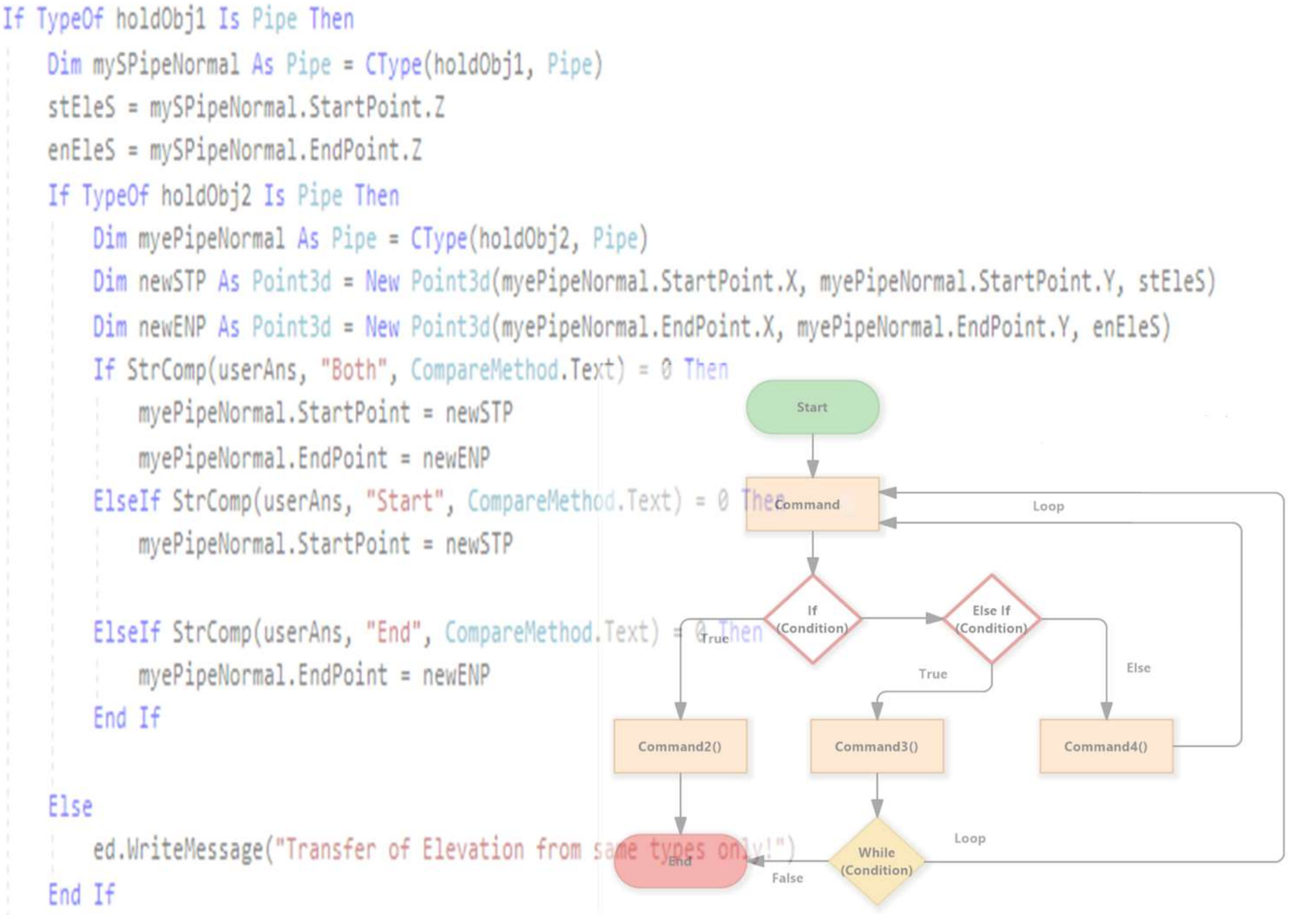
Provisioning for future scale-up

# Create

**Decomposition**

Breaking up a complex problem into manageable chunks

# Create



```
If TypeOf holdObj1 Is Pipe Then
    Dim mySPipeNormal As Pipe = CType(holdObj1, Pipe)
    stEleS = mySPipeNormal.StartPoint.Z
    enEleS = mySPipeNormal.EndPoint.Z
    If TypeOf holdObj2 Is Pipe Then
        Dim myePipeNormal As Pipe = CType(holdObj2, Pipe)
        Dim newSTP As Point3d = New Point3d(myePipeNormal.StartPoint.X, myePipeNormal.StartPoint.Y, stEleS)
        Dim newENP As Point3d = New Point3d(myePipeNormal.EndPoint.X, myePipeNormal.EndPoint.Y, enEleS)
        If StrComp(userAns, "Both", CompareMethod.Text) = 0 Then
            myePipeNormal.StartPoint = newSTP
            myePipeNormal.EndPoint = newENP
        ElseIf StrComp(userAns, "Start", CompareMethod.Text) = 0 Then
            myePipeNormal.StartPoint = newSTP

        ElseIf StrComp(userAns, "End", CompareMethod.Text) = 0 Then
            myePipeNormal.EndPoint = newENP
        End If

    Else
        ed.WriteMessage("Transfer of Elevation from same types only!")
    End If
```

**Charting the flow**

High-level data flow, decision juncture & branching chart

# Create

Capturing / Trapping Errors is another integral part of programming. Without proper error handing, the program could crash at the smallest error, can corrupt databases, or simply not work.

- Pre-Empt – prevent errors from happening by anticipating them and take necessary measures
- Capture – trap the error as it happens, and take the next action based on the type of error.

## Error Trapping

A minimum viable product

A no-frills, non-finished, bare minimum product to see;

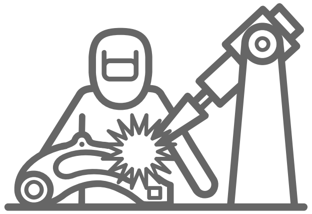
- Flaws in the logic
- Flaws in the process
- Flaws in the structure
- The behavior of various parameters
- Basic error trapping & triggers
- Internal testing

## Prototype/MVP

A minimum viable product

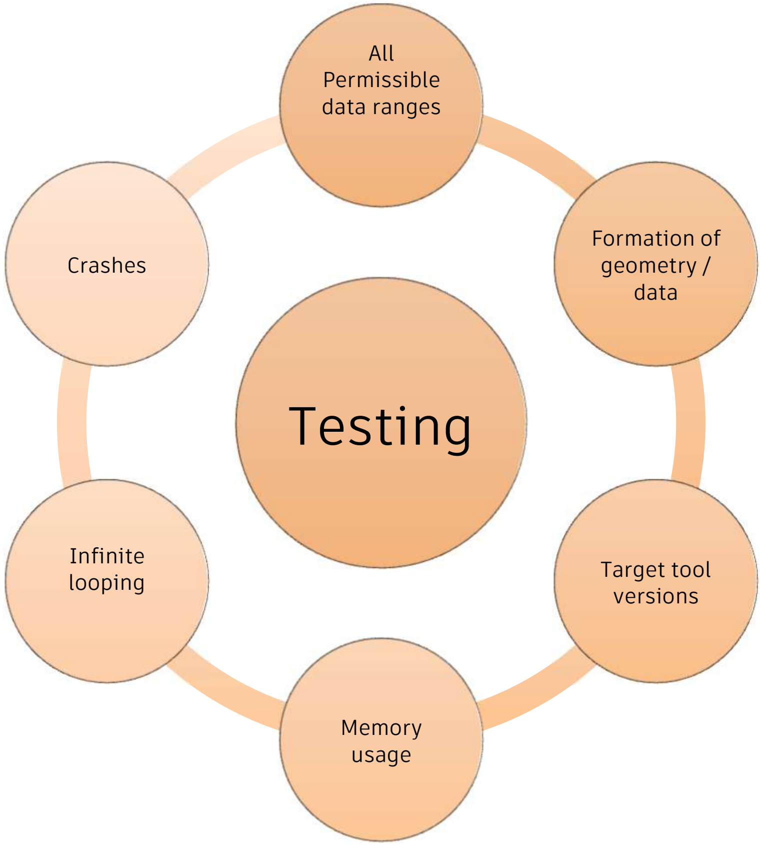
# Test & Debug

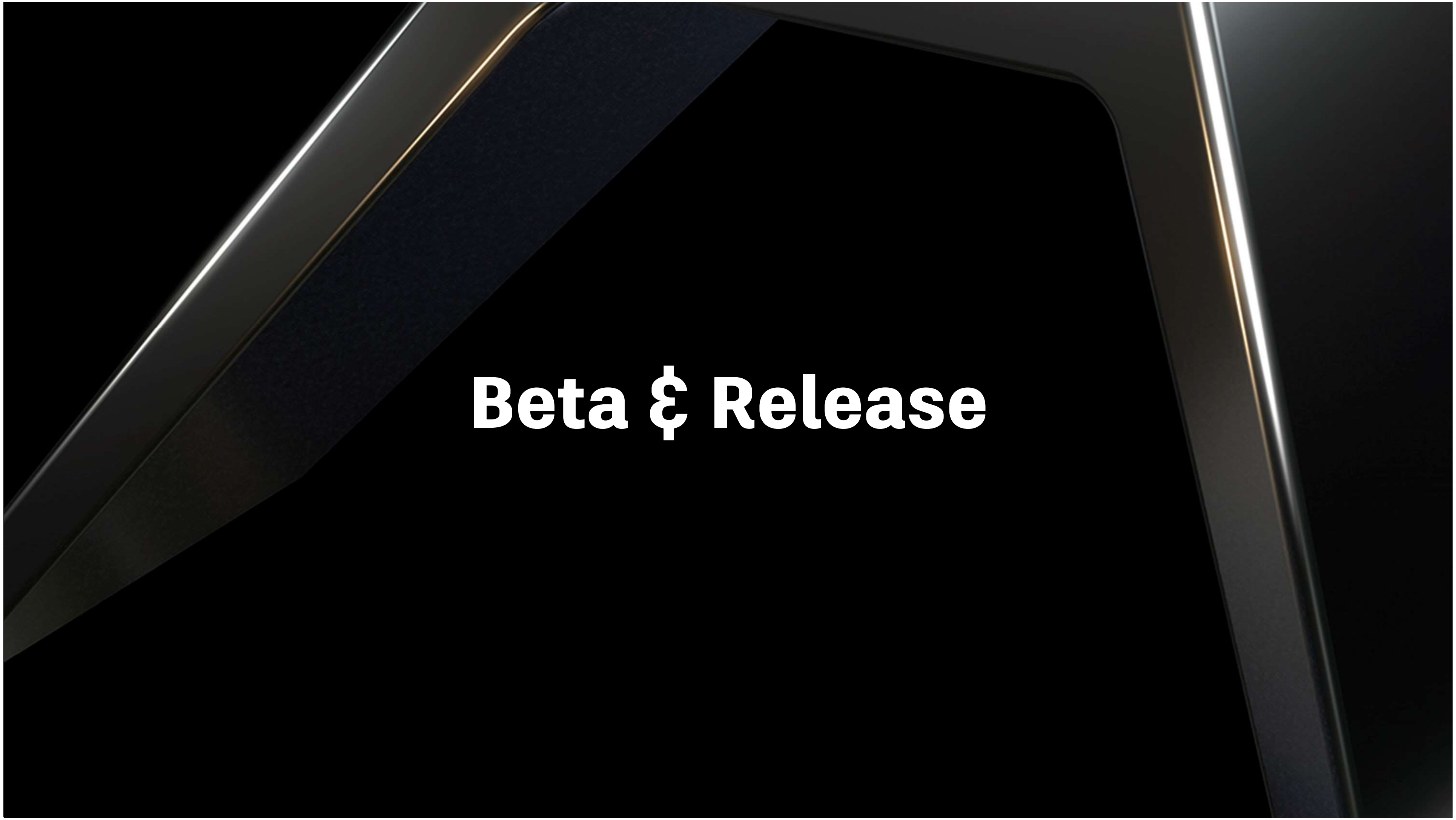
# Testing / Debugging



**Test / Debug**

Testing out the tool & de-bugging the obvious shortcomings, flaws etc





# Beta & Release

# Beta & Release

## Beta Testing

Initial release to domain experts, only to find our the real-world behavior of the tool

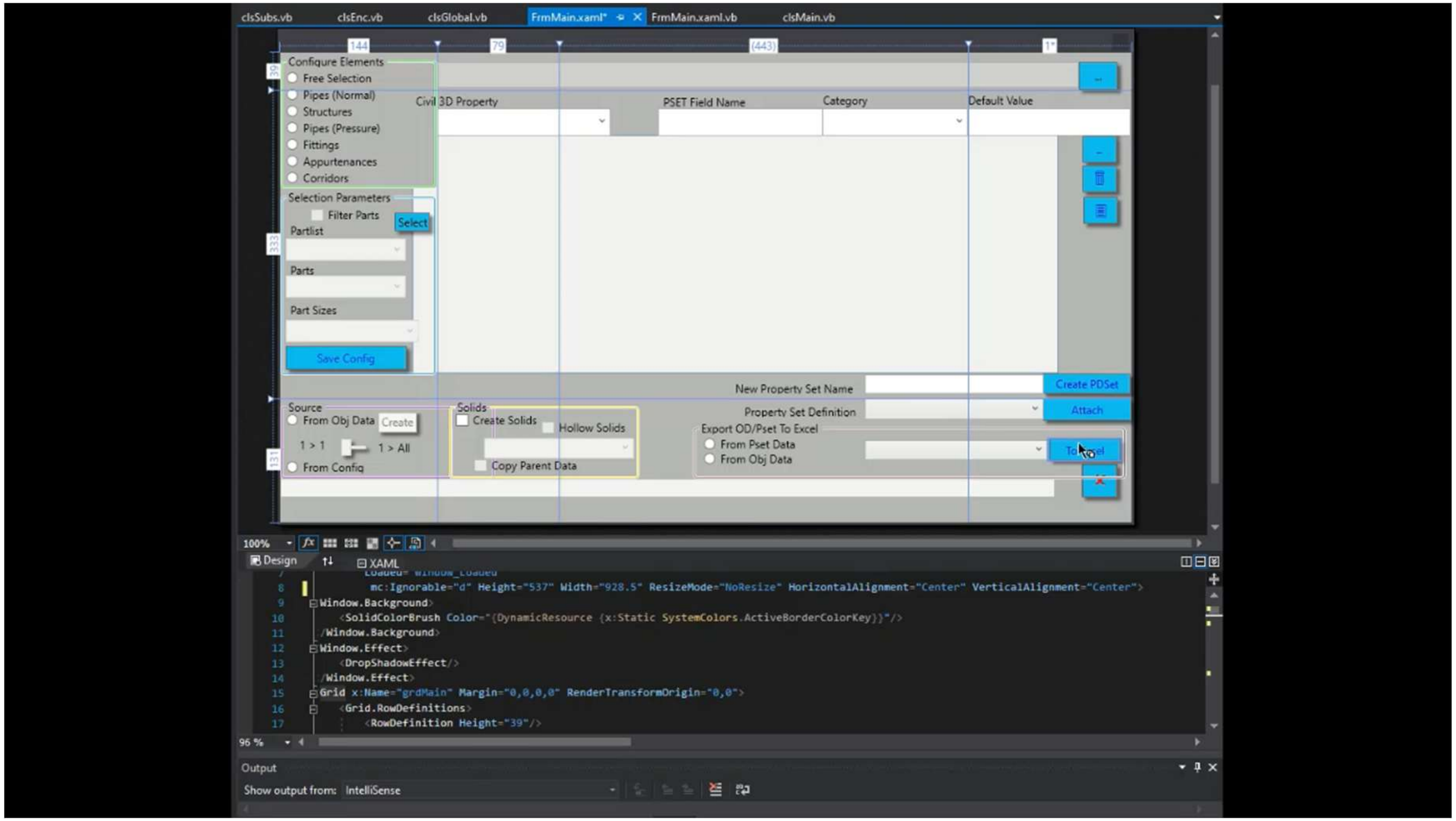
- Version each Beta release – 0.1, 0.2.
- Ensure it works as intended – All computers, environments
- Collect systematic feedback
- Not all feedback can go into the first release.
- Valid comments that is not implemented must go in a future version.

## Release

Tool released to intended audience

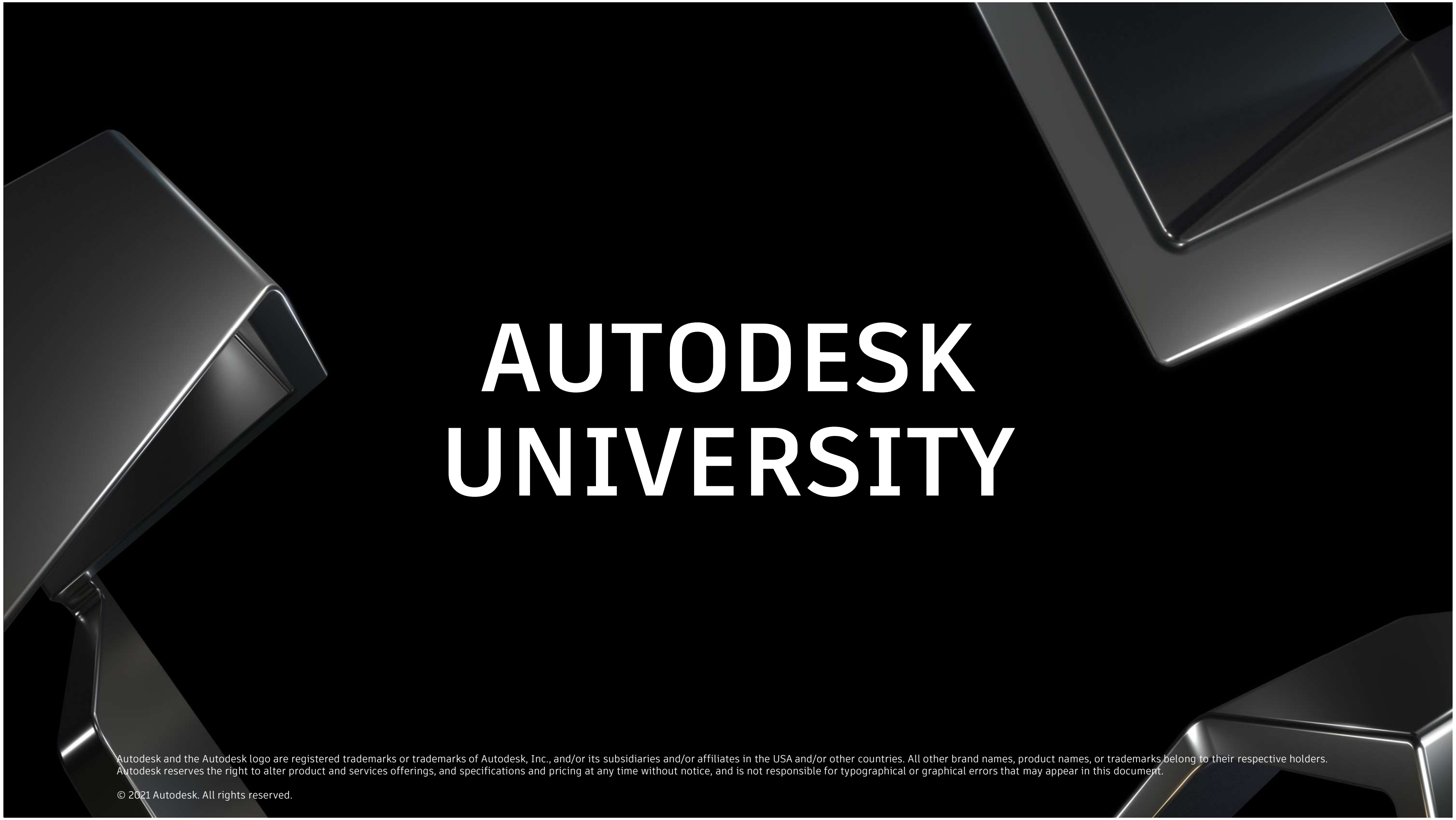
# Quick Demo
# Vb.Net Plugin for C3D

# Demo – Property Data Set Manager



```
        Loaded="Window_Loaded"
        mc:Ignorable="d" Height="537" Width="928.5" ResizeMode="NoResize" HorizontalAlignment="Center" VerticalAlignment="Center">
<Window.Background>
    <SolidColorBrush Color="{DynamicResource {x:Static SystemColors.ActiveBorderColorKey}}"/>
</Window.Background>
<Window.Effect>
    <DropShadowEffect/>
</Window.Effect>
<Grid x:Name="grdMain" Margin="0,0,0,0" RenderTransformOrigin="0,0">
    <Grid.RowDefinitions>
        <RowDefinition Height="39"/>
```

Show output from: IntelliSense



AUTODESK
UNIVERSITY