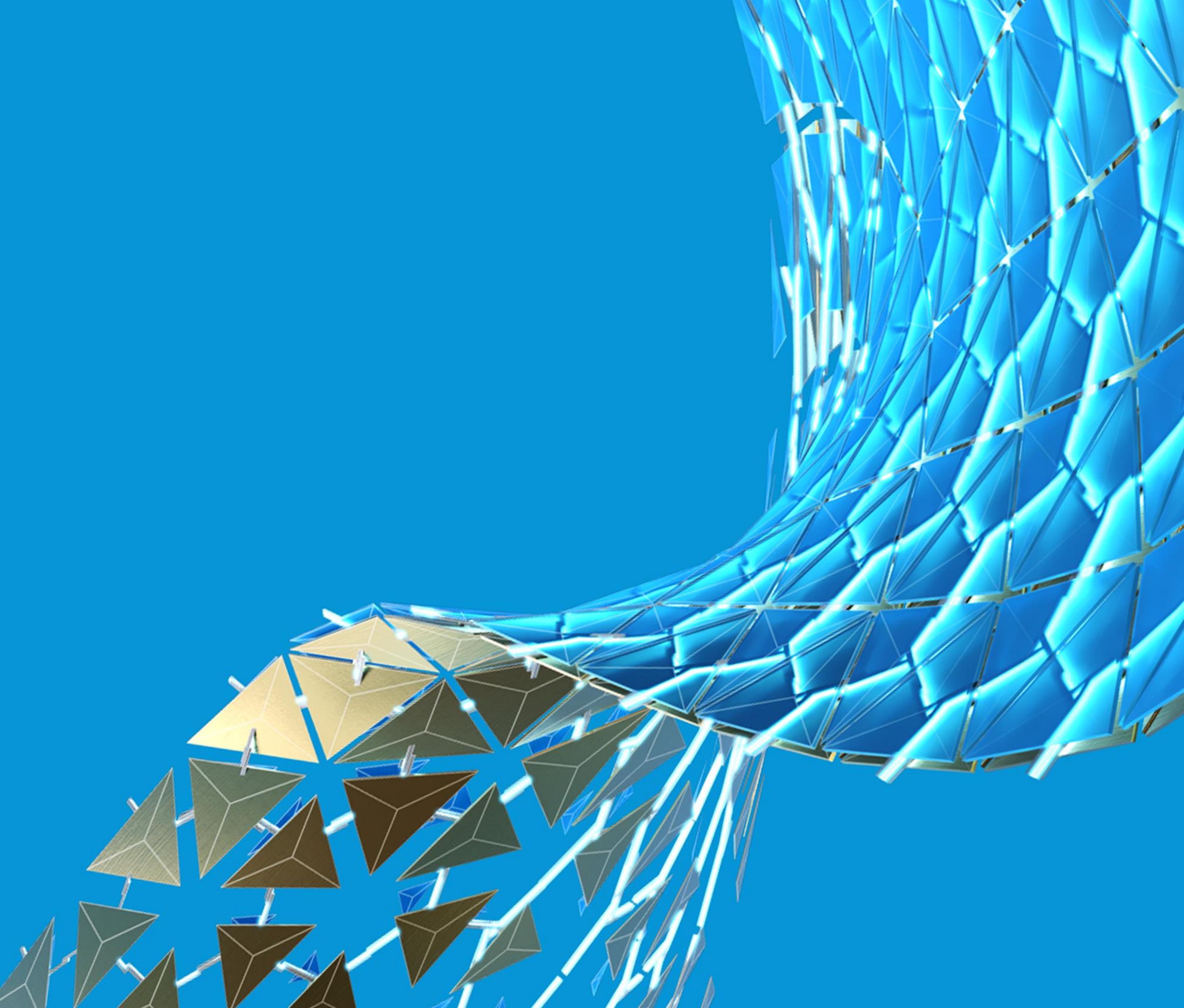


Vom FDU Asset zum Fertigungsteil per iLogic und .net

Pascale Brassat

Dipl.-Ing.(FH) | www.claudiuspeters.com

Hallo



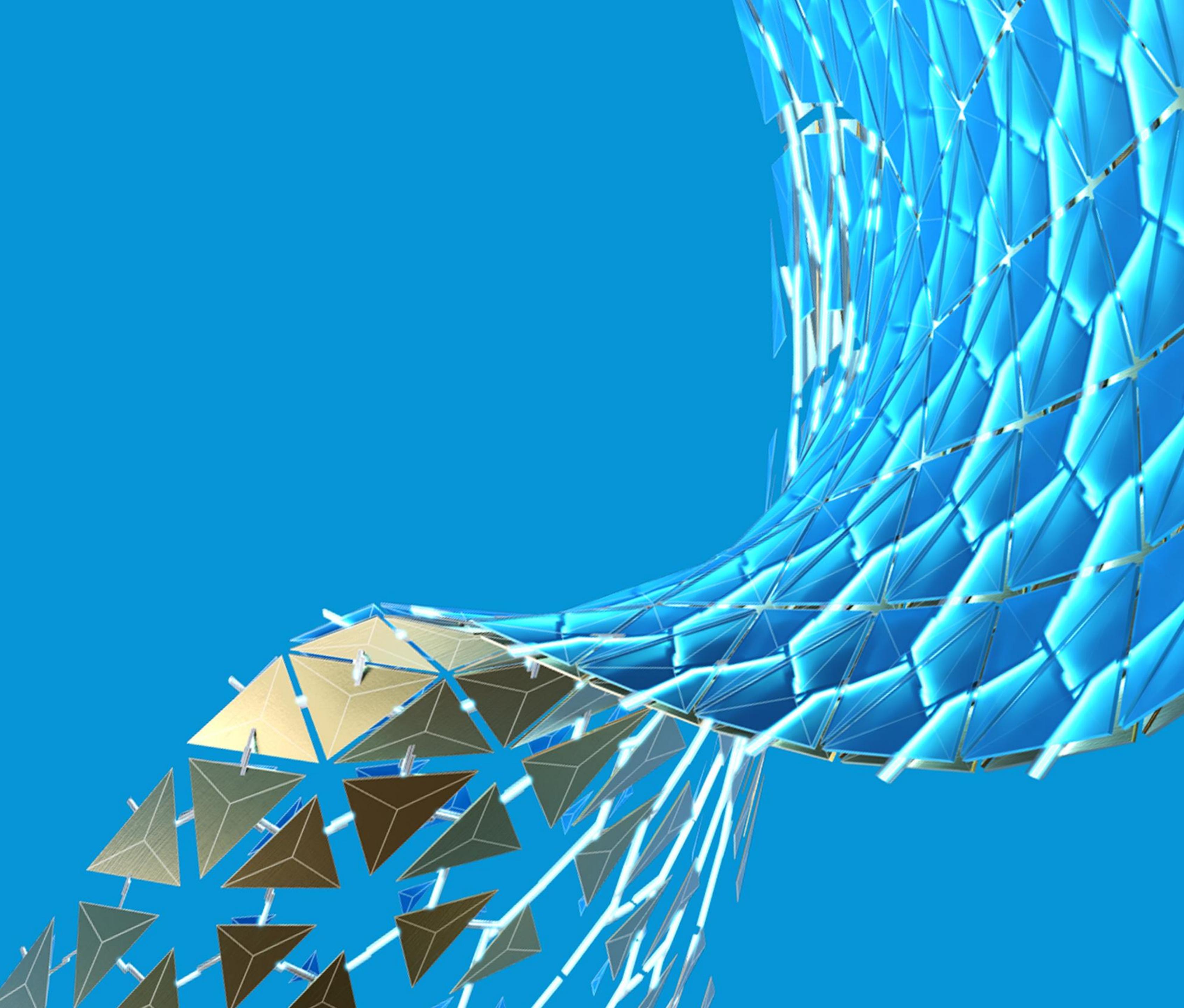


Informationen zur Referentin/ zum Referenten

Pascale Brassat

**Abschluss: Dipl.-Ing.(FH) Fachrichtung
Maschinenbauinformatik. Seit 2006 bei Claudio
Peters Projects GmbH beschäftigt. Zunächst in der
Konstruktion und seit 2011 in der IT Abteilung.
Pascale betreut seit 2008 die Autodesk Produkte im
Unternehmen und beschäftigt sich am meisten mit
den Autodesk Produkten Inventor, AutoCAD
Mechanical und Vault Professional.**

Kurze Info



neu eingefügte Segmente nicht zwingend auch 100mm Durchmesser besitzen. Das bedeutet, ein Editieren des neuen Segments ist notwendig.

Seite 16

Seite 17

Seite 18

Seite 19

Seite 20

Beim Arbeiten über die Grenzen des eigenen Arbeitsplatzes hinaus, muss sichergestellt sein, dass die **Geometriedaten** immer lokal bei dem jeweiligen Anwender liegt.

Animation funktioniert das mit dem Vererben nicht.

Das wird im Normalfall aber leicht bemerkt. Funktioniert das nicht mit der Vererbung nicht, ist es vermutlich die Datei oder ist nicht aktuell, ganz simpel.

Wir haben noch eine Besonderheit bei unseren Connexoren:

Da die PDU bei den Connexoren nicht die Richtung unterscheidet (Bei Kanälen, Röhren mit Abzweigungen etc. nicht un interessant), helfen wir uns auf Farben der Anteile zu beziehen.

Blau = Einlauf
Rot = Auslauf
Grün = Generelle Verbindungsleitung ohne bestimmte Richtung

Dadurch ist es leichter für den Anwender zu erkennen, welches Bauteil wo genau eingesetzt werden muss in einem fortlaufenden Kanal o.ä.

Hervigabele erstellen
Wie haben wir nun alle erreicht.
Wir haben die benötigte Bibliothek am PDU Asset erstellt.
Das heißt eine Planung mit vereinfachten Teilen ist nun möglich und funktioniert auch ziemlich schnell.
Doch das Ziel ist es, fertigungsgemäße Modelle zu haben in der jeweils korrekten Ausprägung.
Nun wird also ein Modell aufgebaut, das in der Geometrie dem Asset entspricht, aber von Anspruch her so modelliert wird, wie in Inventor normierweise ein Fertigungsmodell aufgebaut wird.
Das heißt Blechteile z.B. werden auch als Blechteile erzeugt um davon Abwicklungen zu entstehen.
iLogic im Baugruppen
Zunächst einmal wird diese Baugruppe für eine spezifische Baugruppe und Ausprägung modelliert.
Ist diese Variante fertig und sauber modelliert, wird damit begonnen die Logic Regeln zu entstehen.
Wir haben immer in der obersten Baugruppe eine Regel mit dem Namen "Durchgang" also **Bearbeitungsregel** oder Größen.
Diese Regel startet dann die Übertragung der Parameter an die notwendigen Stellen.
Das heißt die Regel wird aktiviert im Falle einer Parameteränderung der Key Parameter. In diesem Fall die Parameter die später von dem Asset an das detaillierte Modell übertragen werden sollen.
Die Baugruppen Regel überträgt dabei Parameterwerte an die untergeordneten Teile oder Unterbaugruppen, die dann wieder ihrerseits auf die Parameter reagieren. Damit hat jedes untergeordnete Element für sich die Kontrolle über die darin enthaltenen Parameter.
Also es wird nicht von oben herab ein Teil geprägt, sondern nur etwas übertragen und dann im jeweiligen Unterelement wird darauf reagiert.
Da in der Baugruppe die Bauteile später automatisch neue Namen bekommen müssen, also auch alle untergeordneten Teile und Baugruppen, ist es wichtig, dass die Vererbung der Informationen innerhalb von Inventor und Solid

Aber mit anderen Worten, wenn man in iLogic ein untergeordnetes Element über den Datennamen oder anfanglichen, veränderlichen Element-Namen anspricht, dann funktioniert die Logic Regel nur ein einziges Mal und dann nicht mehr, weil sich der Name verändert.
Aus dem Grund haben wir alle Elemente die im Logic angesprochen werden mit einem vorangestellten **Assetname**.

Die anderen Elemente sind egal, weil sie z.B. doppelt vorkommen etc.

Segment mit der Regel beim Ändern der benannten Parameter startet

Segment für die Übertragung der Parameter an die Unterelemente

Ende und Aktualisierung des Inventor Dokumentes

Und jetzt kommen wir zur nächsten Überlegung. Die Parameter sollen aus dem Asset übertragen werden. Aber sicher soll dies Arbeit machen, welche Fertigungsetat durchgeführt.
Es soll später möglichst automatisiert erfolgen.
Als unser Bauteile erhalten **Teileid**. Diese Nummer sind eindeutig. Also ein Bauteil hat eine **Teileid** hat der Nutzer die **Teileid**, hilft er auch das Bauteil.
Im Asset wird bei uns in den **Properties** ein Hinweis auf das Fertigungsmodell gesetzt:
(Im Feld "Part Number")

iLogic (Part Number) Properties

General Summary Print Status Color Save Document Print
Version: 0 Autodesk Inventor Professional Asset
Name: Bauteile
Part Number: 1000-1000
Stock Number:
Category:
Project:
Region: Global
Author:
Comments:
Last Saved: 10/10/2018 10:10:10 AM
Saved By: 1000-1000
File Type: Part
Notes:
Status: Active

Das **Teileid** steht nur als Identifikation, dass es sich bei der eingegebenen **Teileid** wirklich um ein Asset handelt. Dieser Hinweis auf das Fertigungsmodell handelt.

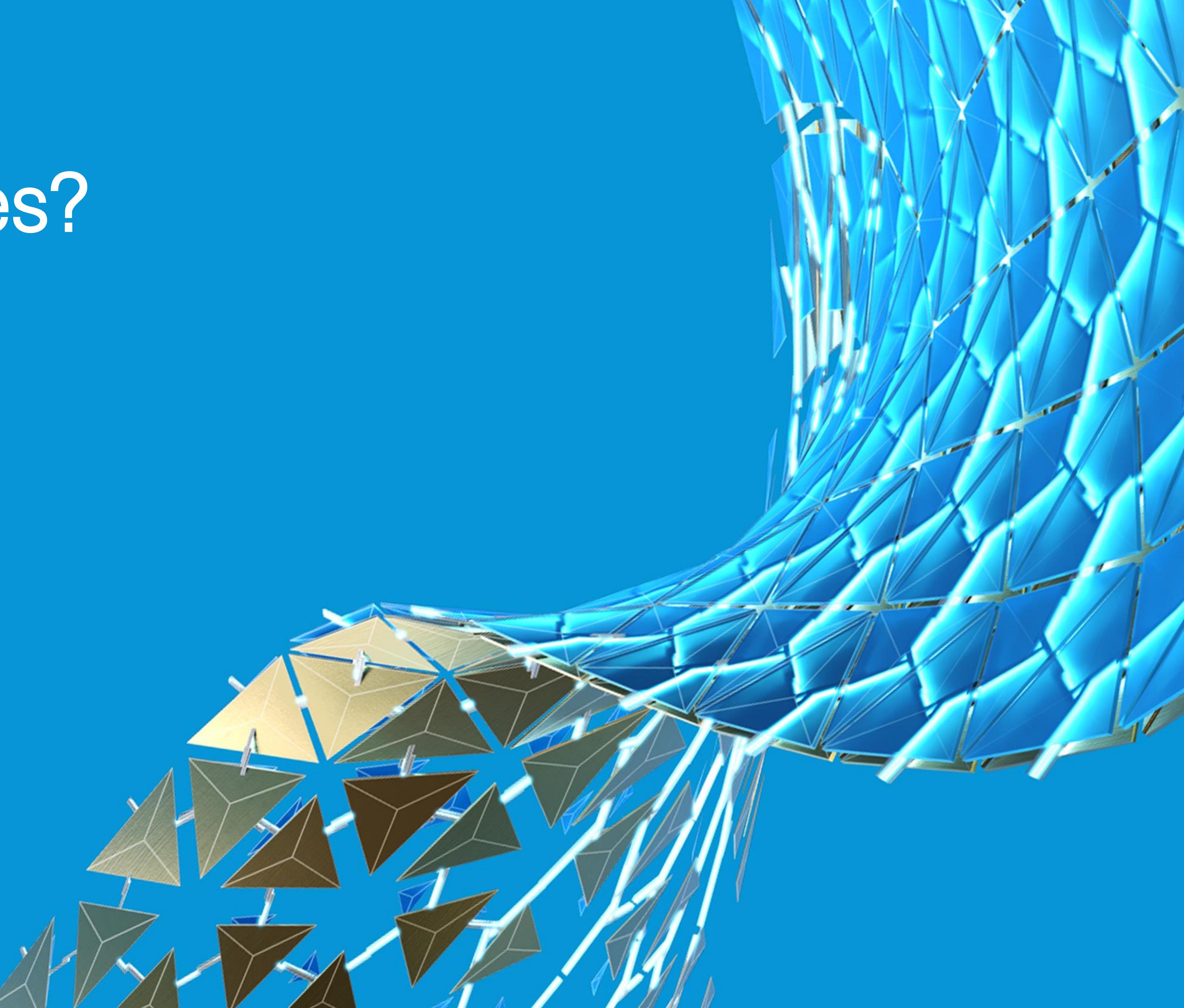
Wir setzen bei uns Inventor Professional ein und das machen wir uns an dieser Stelle zu nutzen.

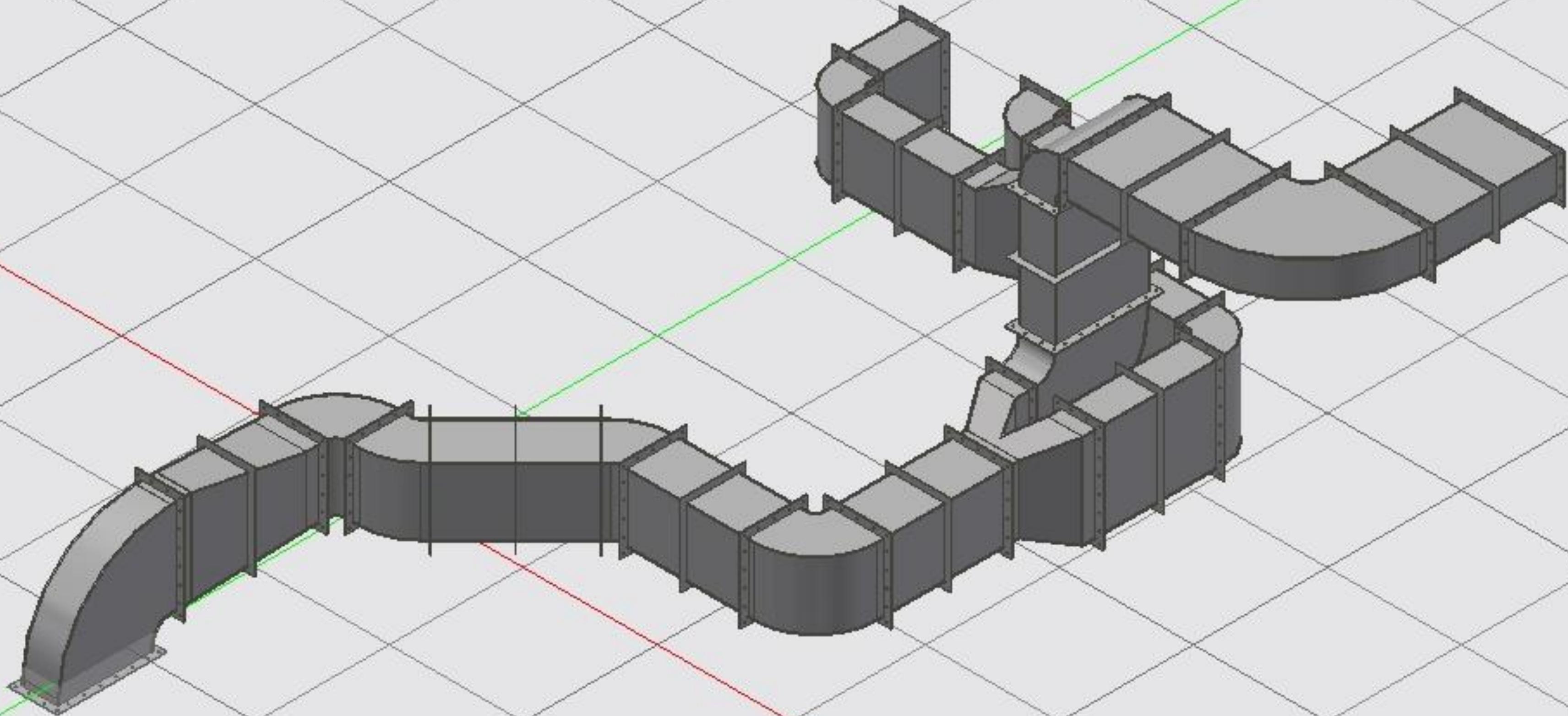
Weitere Informationen im Handout

Zu dieser Session gibt es auch ein Handout. Dieses enthält detailliertere Informationen, welche in der Kürze der Zeit in dieser Session nicht dargestellt werden konnten.

Diese Session erweckt hoffentlich den Wunsch nach "Mehr"

Worum geht es?





Beispiel Belüftungskanal

Anhand der Belüftungskanäle kann prima demonstriert werden, was die Kombination aus Inventor, iLogic, FDU und .net bewirken kann

Was will ich in dieser Session zeigen?

iLOGIC

Möglichkeiten von iLogic in Inventor darstellen.

.NET

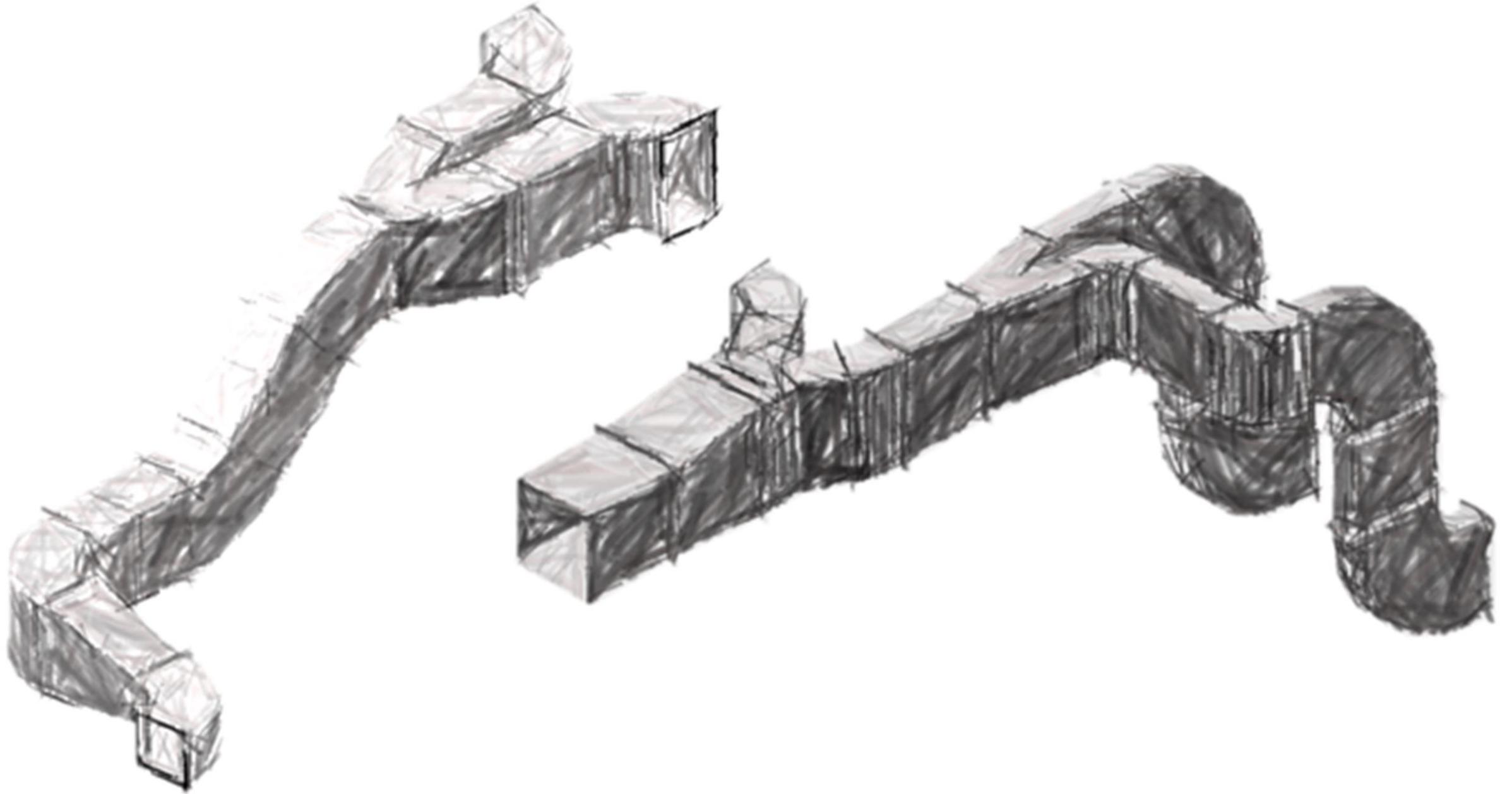
Sehr grob darstellen, was mit Hilfe von .net (im Speziellen vb.net) und der API von Vault und Inventor möglich ist.

FDU

Zeigen, wo die Factory Design Utilities besonders effektiv genutzt werden können.

KOMBINIEREN VON TECHNOLOGIEN

Ein Verständnis dafür geben, warum die Kombination von mehreren Technologien eine besonders effektive Idee sein kann.



Warum das Ganze?

Der Hauptgrund war ganz einfach: Zeitaufwand minimieren

Wir haben bei uns das Problem, dass Änderungen an unseren Anlagenplanungen immer bis kurz vor Abgabetermin erfolgen.

Daher sind wir besonders an schnellen Lösungen interessiert, die es uns ermöglichen erst im letzten Augenblick die benötigten Unterlagen besonders effektiv zu erzeugen.

Wie lief bisher eine Planung ab?

Der unten beschriebene Ablauf wiederholte sich je nach Änderung zum Teil mehrfach

- Der Leitungsverlauf wurde im Anlagenmodell grob skizziert
- Dann wurde auf Basis dieser Skizze ein Sweeping erzeugt
- Dieses Sweeping wurde dann in mögliche Komponenten zerlegt
- Diese Komponenten wurden dann als Baugruppe modelliert
(teils auf Basis existierender Teile, teils neu)
- Aus den Teilen wurden dann Zeichnungen erstellt

Viel zu zeitintensiv und eine optimierte Lösung musste her!

Große Zieldefinitionen

ABLAUF BESCHLEUNIGEN

Die erste Aufgabe war, die Prozessabläufe so zu beschleunigen, dass eine Änderung in letzter Minute beherrschbarer wird.

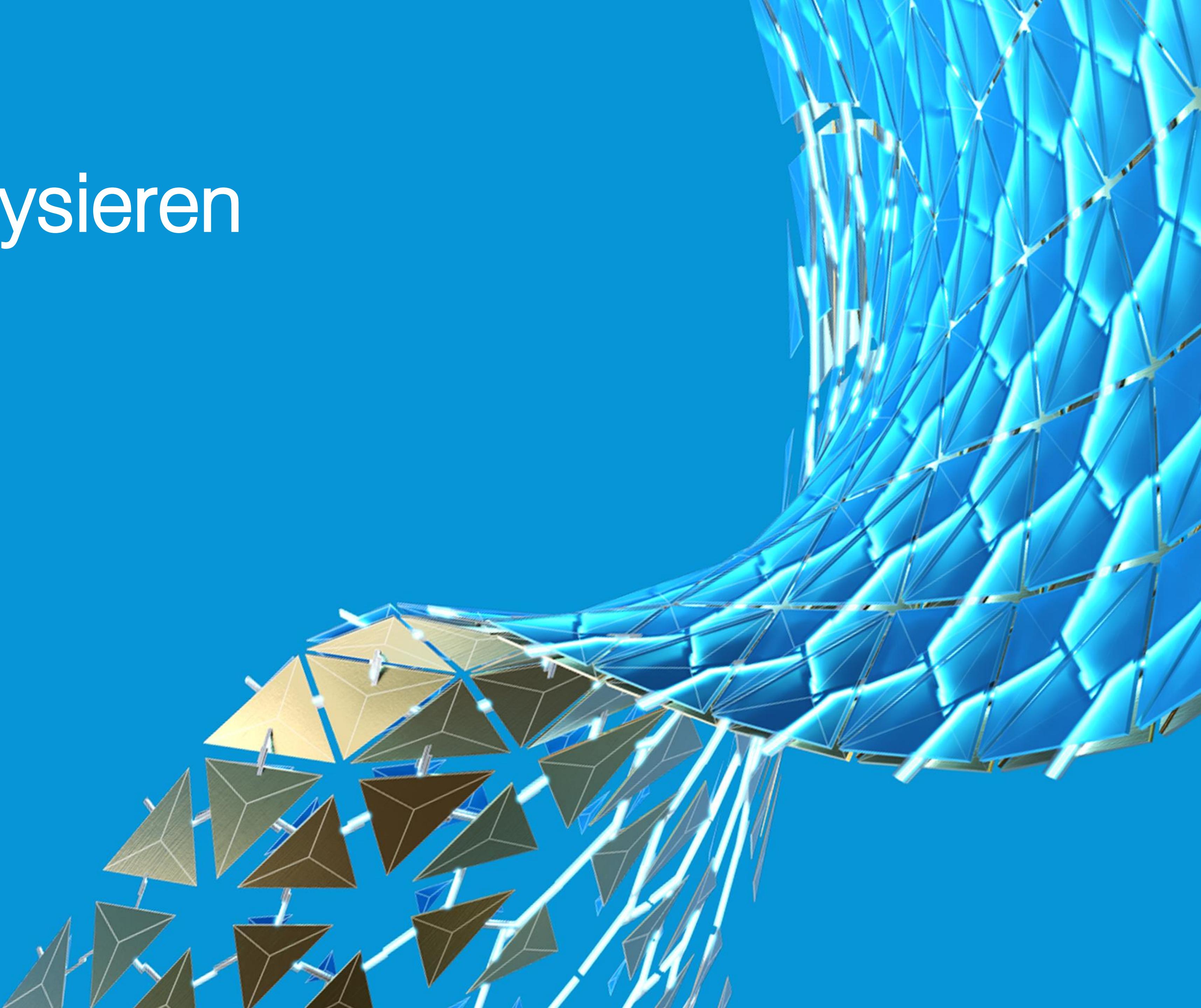
AUF DAS WESENTLICHE KONZENTRIEREN

Alles sollte so einfach wie möglich für den Anwender (oder Anwenderin) ablaufen. Die Arbeitszeit soll genutzt werden, um Leitungen zu planen und abzuarbeiten und nicht um Zeit zu verschwenden weil der Lösungsweg zu komplex geworden ist.

IM IDEALFALL SOLLTE DAS PRODUKT NOCH OPTIMIERT WERDEN

Eine Möglichkeit sich mit einem Ablauf um ein Produkt herum zu beschäftigen, bietet auch immer die Option das Produkt selbst zu optimieren.

Problem analysieren



Basis Idee

Idealerweise sollte es möglich sein, mit Hilfe eines Baukastensystems alle möglichen Belüftungskanäle zu planen.

Da es bereits jede Menge geplanter Leitungen im 2D und 3D Bereich gibt, war die erste Aufgabe zu schauen, welche Teile/Segmente notwendig sind für die Planung.

Bei dieser Analyse gab es dann auch die Möglichkeit mit allen betroffenen Abteilungen zu sprechen, um deren Wünsche zu berücksichtigen (Einheitliche Flansche, veränderte Abmessungen, Überlegungen bezüglich Strömungstechnik etc.)





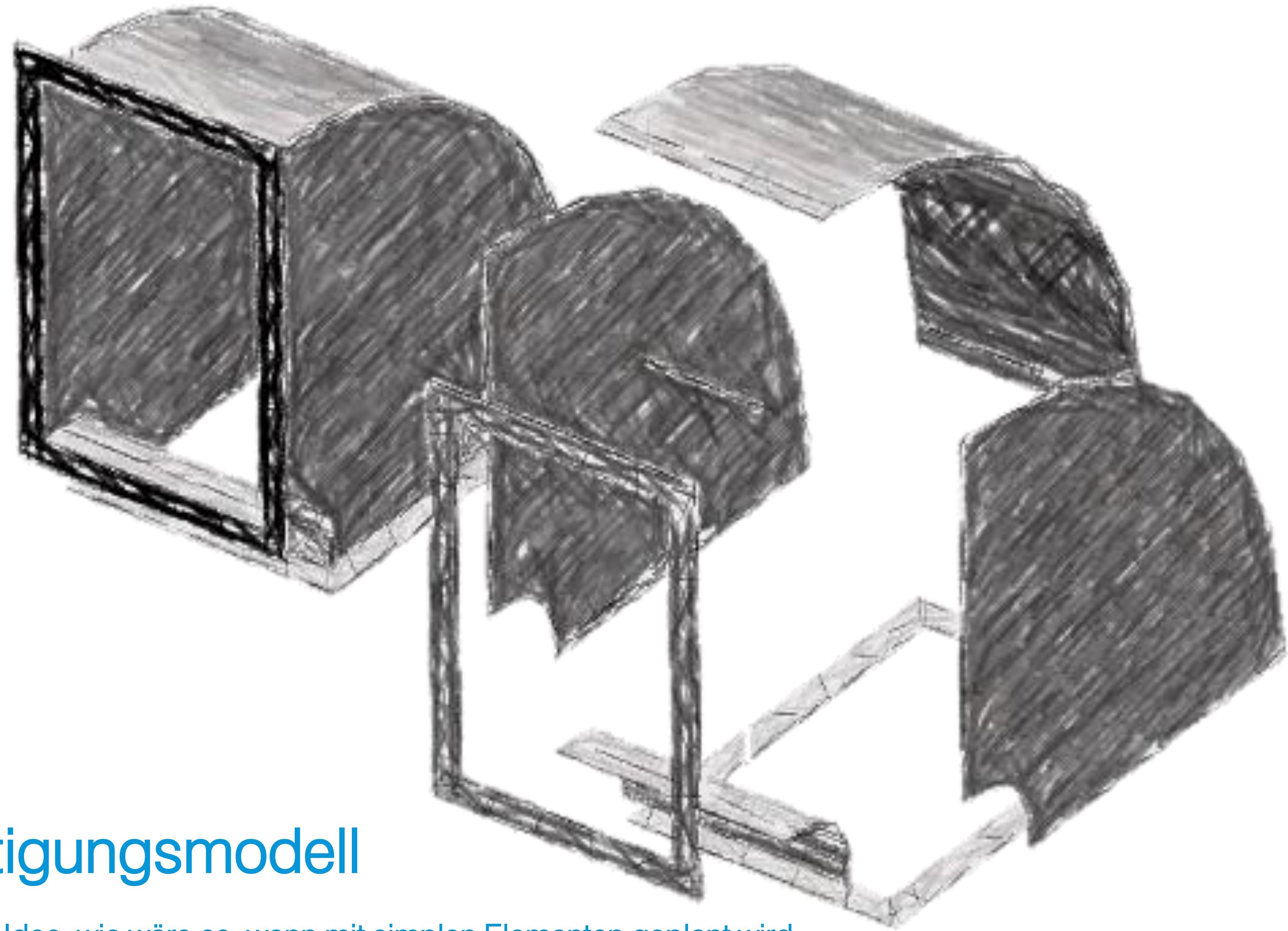
Benötigte Komponenten

Insgesamt wurden 9 Komponenten identifiziert, um alle möglichen Verläufe abzubilden

4 Basisformen + 1 (aufgrund der Connectoren) und 4 weitere Teile (z.B. Stützen)

Jede Menge Varianten

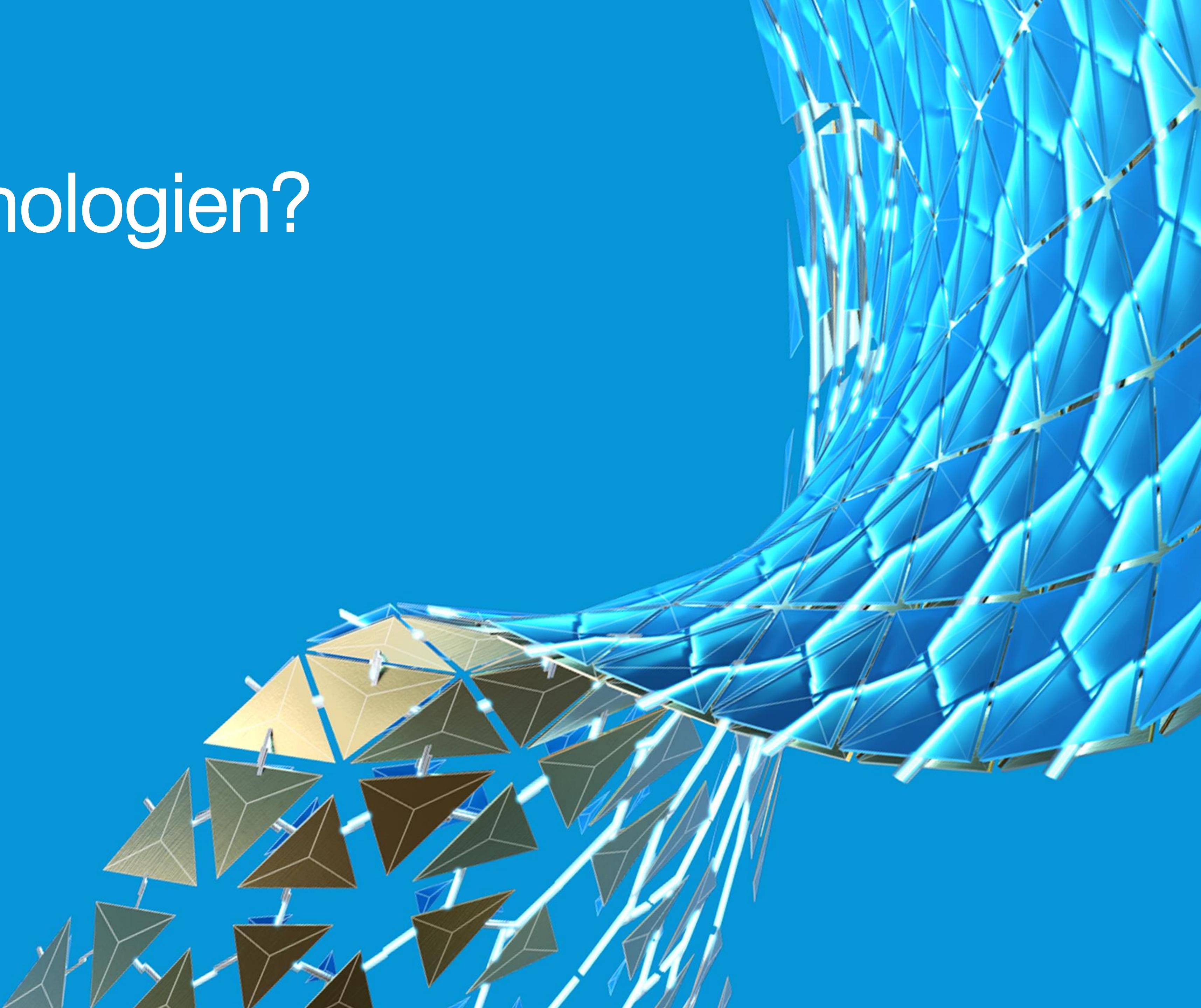
Durch die Kombination von mehreren Parametern mit jeweils einer großen Menge möglicher Werte ergeben sich viele Varianten.



Vom Asset zum Fertigungsmodell

Bei der Analyse des Problems kam die Idee, wie wäre es, wenn mit simplen Elementen geplant wird (schnell und performant) und anschließend daraus Fertigungsmodelle entstehen???

Welche Technologien?



Bedingungen

LEICHT PLANBAR

Die Technologie soll von vielen Nutzern/-innen anwendbar sein, ohne aufwendige Schulungsmaßnahmen.

VIELE VARIANTEN ABBILDBAR

Es gibt jede Menge Varianten, diese sollen alle abbildbar sein.

Pflege soll nicht von allen betrieben werden, sondern nur von bestimmten Personen.

WENIG ARBEITSSCHRITTE

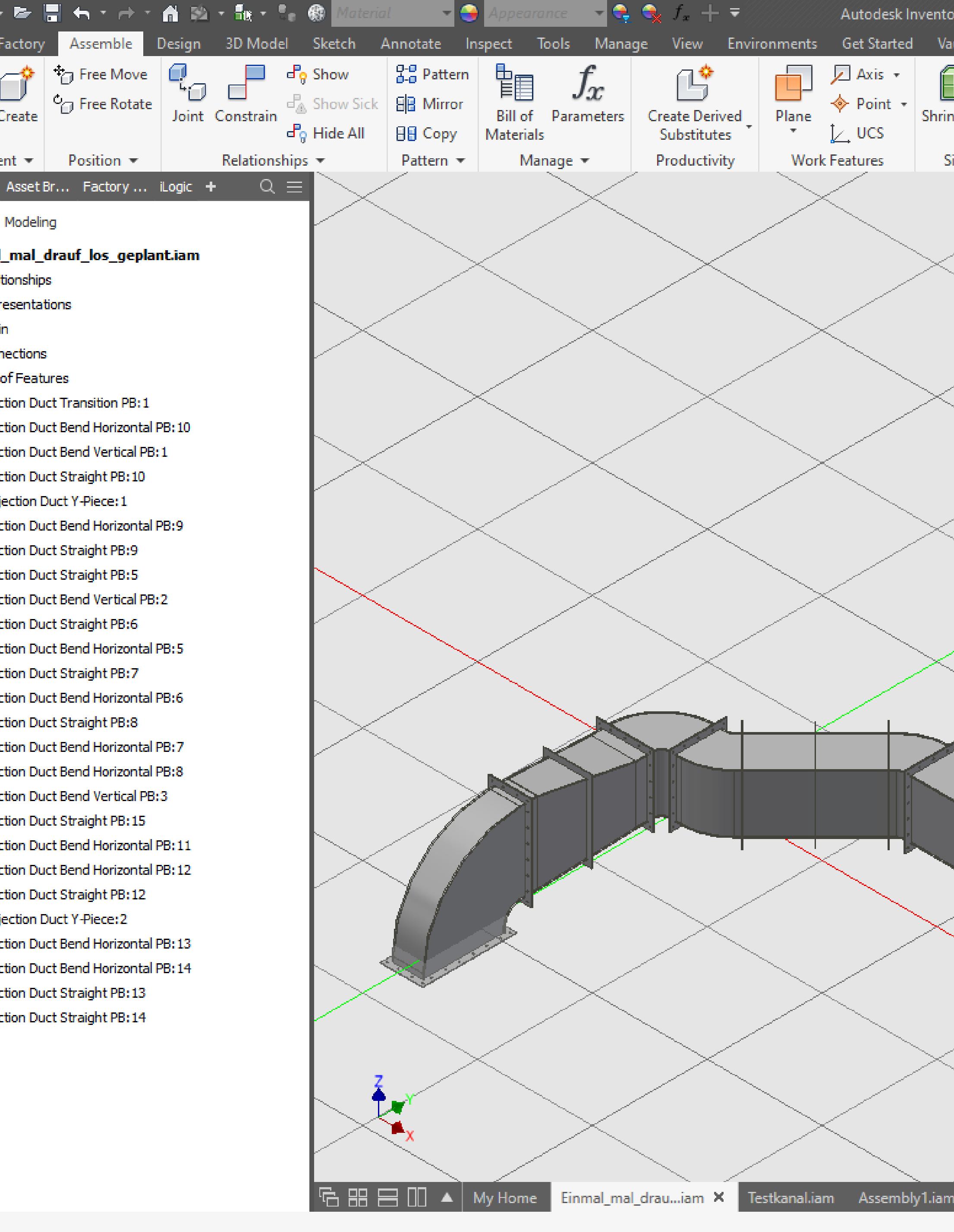
Es sollen möglichst weniger Arbeitsschritte zum Ziel führen als bisher.

Vor allem der Punkt: Vom Asset zum Fertigungsmodell findet nur Akzeptanz, wenn er mit wenig Aufwand funktioniert.

Inventor

Inventor ist als Basis für alle weiteren Überlegungen gesetzt.
Es ist das Standard Programm im CAD Bereich unserer Konstruktionsabteilungen und wird daher auch hier verwendet.

- 3D und 2D möglich
- Parametrik ist sehr vielseitig
- Anbindung an Vault
- API vorhanden für ergänzenden Dinge die nicht im Standard enthalten sind
- Programm sehr vielfältig nutzbar
- FDU als Ergänzung vorhanden



iLogic

iLogic ist eine vollwertige Programmierumgebung um z.B. komplexe parametrische Ideen umzusetzen.

- Für VBA Anwender sehr leicht zu erlernen
- Eine breite Palette an vorgefertigten Elementen
- Event gesteuert
- Vielseitig nutzbar (nicht nur Parametrik, sondern auch Steuerung der Inventor Instanz etc.)
- Es lassen sich auch Forms erzeugen
- Erweiterbar durch externe DLL's

The screenshot shows the 'Edit Rule: Flange_Holes' dialog box. On the left, there's a 'Snippets' panel with categories like System, Custom, Parameters, Features, Components (classic), etc. The main area displays a hierarchical tree of model components: B Injection Duct Y-Piece, Model Parameters, Reference Parameters, User Parameters, View: Master, Sketch1, Work Plane1, Extrusion1, Extrusion2, and a few other nodes. Below the tree is a toolbar with various icons. The right side is a large text area containing the following VBA-like script:

```
Parameter("dim_1_y2") = Wert3 / 2

'Horizontale Lochteilung Flansch 2

Wert1 = Parameter("dim_W2") + (Parameter("dim_Flange_2_Width")

Parameter("d83") = Math.Floor((Wert1 / dim_dist_Holes))

Wert3 = Wert1 - ((Parameter("d83") - 1) * dim_dist_Holes)

Parameter("dim_2_y") = Wert3 / 2

'Vertikale Lochteilung Flansch 2

Wert1 = Parameter("dim_H2") + (Parameter("dim_Flange_2_Width"

Parameter("d86") = Math.Floor((Wert1 / dim_dist_Holes))

Wert3 = Wert1 - ((Parameter("d86") - 1) * dim_dist_Holes)

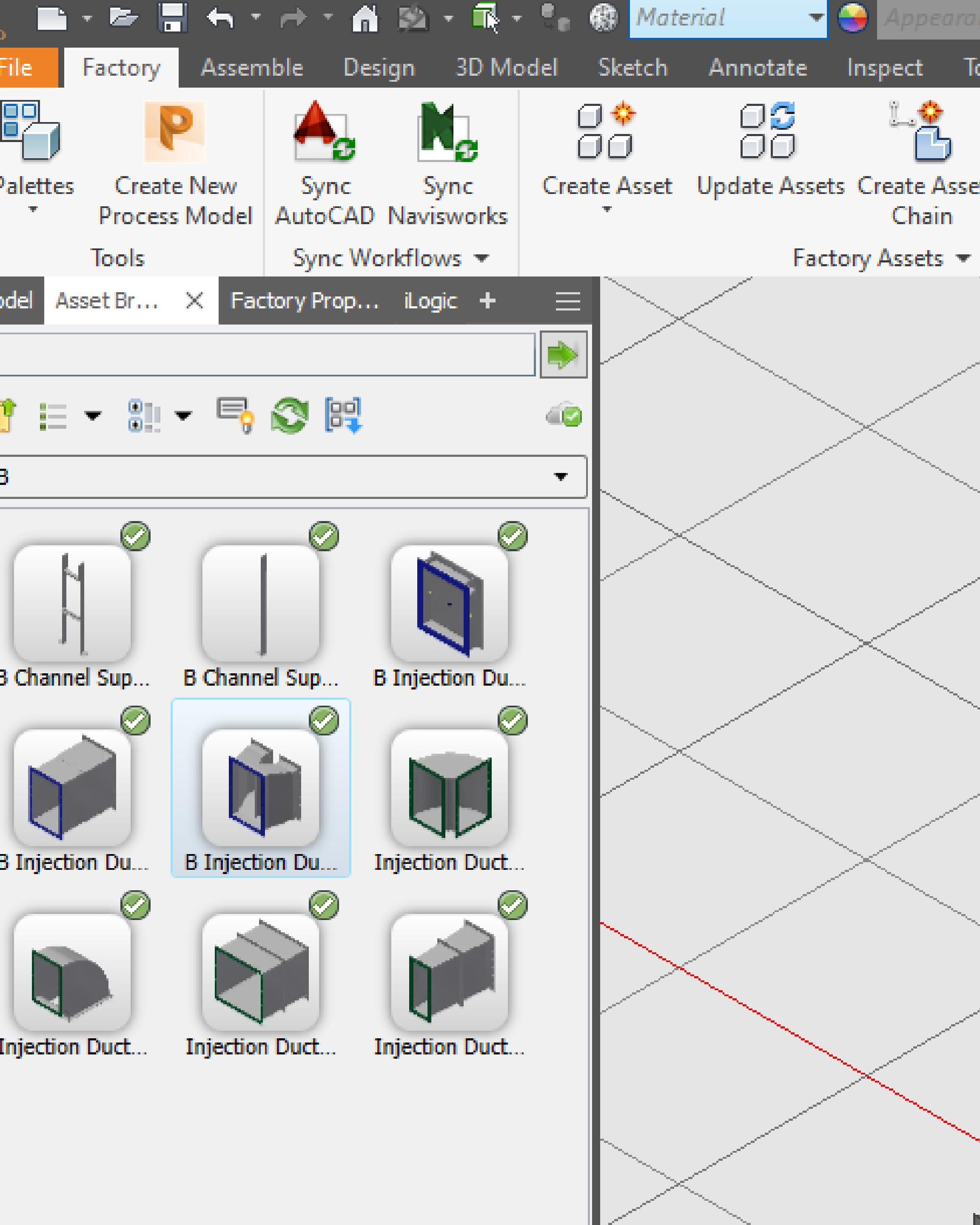
Parameter("dim_2_y2") = Wert3/2

iLogicVb.UpdateWhenDone = True
```

At the bottom, there are buttons for Log Level (Info, Detailed Trace checked), and a status bar with 'Detailed Trace' checked.

Die Factory Design Utilities bringen eine Menge interessanter zusätzlicher Funktionen in Inventor. Anders als der Name vermuten lässt, nicht nur für die Fabrikplanung

- Leichte Erstellung von Bibliotheken
- Die FDU vereinfacht viele Prozesse (z.B. Dateibenennung beim Verändern)
- Connectoren zum Verbinden von Assets
- Vererbung von Parameterwerten möglich
- Vielseitig einsetzbar



Vault Professional

Mit Vault Professional lassen sich Dokumente verwalten,
Stücklisten erzeugen etc.

- Dateireferenzen lassen sich schnell nachvollziehen
- Verschiedene Dateiversionen lassen sich verwalten
- Dateien lassen sich schnell suchen und finden
- Zugriffssteuerung
- Eindeutige Dateinamen lassen sich umsetzen

The screenshot shows the Autodesk Vault Professional interface with the 'Home' workspace selected. The left sidebar lists navigation options: 'Home', 'Project Explorer', 'Factory Asset', 'Factory Layout', and 'Item Master'. The main area displays a hierarchical file tree under '150700-01' and a detailed list of files with their states.

File Tree:

- 150700-01
 - 625
 - 626
 - 629
 - 631
 - 634
 - 637
 - 638
 - 639
 - 640
 - 641
 - 643
 - 645
 - 650
 - 652
 - 652_0001
 - 652_0002
 - 652_0003
 - 652_0005
 - 652_0021
 - 652_0100
 - 150700-01

File List:

Name	State
150700-01.dwg	RELEASED
150700-01.iam	RELEASED
150700-01-Pos10.ipt	RELEASED
150700-01-Pos15.ipt	RELEASED
150700-01-Pos50.ipt	RELEASED
150700-01-Pos60.ipt	RELEASED
150700-01-Pos90.ipt	RELEASED
150700-01-Pos100.ipt	RELEASED
150700-01-Pos130.ipt	RELEASED
150700-01-Pos140.ipt	RELEASED
150700-01-Pos160.ipt	RELEASED
150700-01-Pos165.ipt	RELEASED
150700-01-Pos170.ipt	RELEASED
150700-01-Pos180.ipt	RELEASED
150700-01-Pos190.ipt	RELEASED
150700-01-Pos200.ipt	RELEASED
150700-01-Pos210.ipt	RELEASED
150700-01-Pos220.ipt	RELEASED
150700-01-Pos235.ipt	RELEASED
150700-01-Pos250.ipt	RELEASED
150700-01-Pos260.ipt	RELEASED

.net Inventor und Vault API

Mit Hilfe der API von Inventor und Vault Professional lassen sich nahezu alle Ideen umsetzen. Fehlt eine Funktion in Inventor? Dann lässt sie sich vermutlich in .net unter Nutzung der API erstellen

- Nahezu keine Grenzen
- Es gibt eine sehr aktive Community
- Viele Ressourcen im Internet verfügbar
- iLogic und .net Code lassen sich kombinieren

```
Dim fileSave As Inventor.FileSaver = Apprentice.FileSaver

Dim New_Name As String = ""
New_Name = System.IO.Path.GetFileNameWithoutExtension(Dok.FullFileName)
New_Name = New_Name.Replace(MatchName, ReplaceName)

Dim Zeit As String = Allgemeine_Funktionen.Zeitstempel()

If Zeitstempel = True Then
    New_Name = System.IO.Path.GetDirectoryName(Dok.FullName).Replace("\\", "\")
Else
    New_Name = System.IO.Path.GetDirectoryName(Dok.FullName).Replace("\\", "\") & "-" & Zeit
End If

Dim NeuerPfad As String = System.IO.Path.GetDirectoryName(Dok.FullName)
NeuerPfad = NeuerPfad & "-" & Zeit

Erstelle_Kopie_2 = New_Name

NeuerName = New_Name

Dim Liste As New ArrayList
Liste.Add(NeuerName)

fileSave.AddFileToSave(Dok, New_Name)

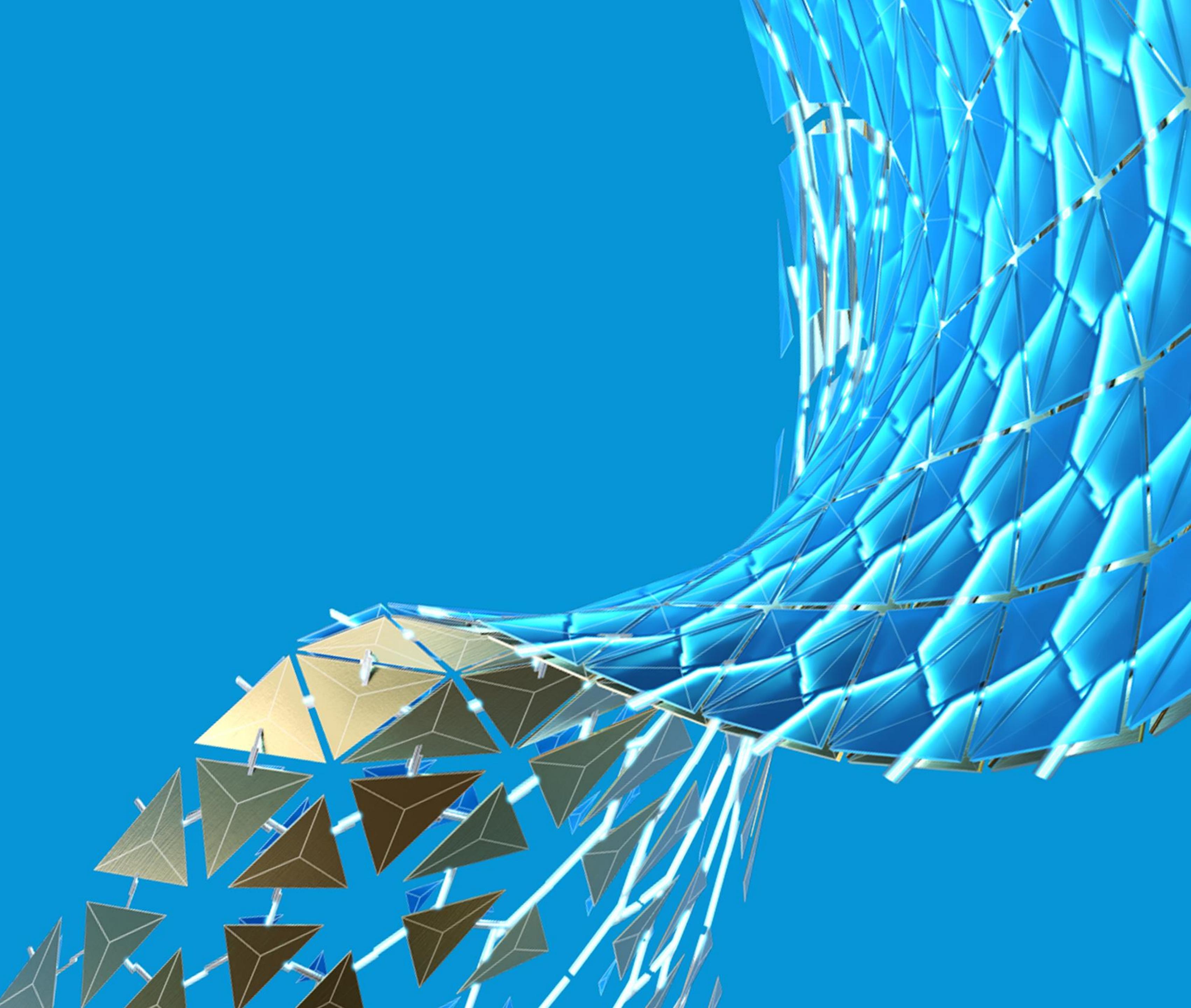
Dim Text As String = ""
Dim i As Integer
Dim Vorschlag As String = ""

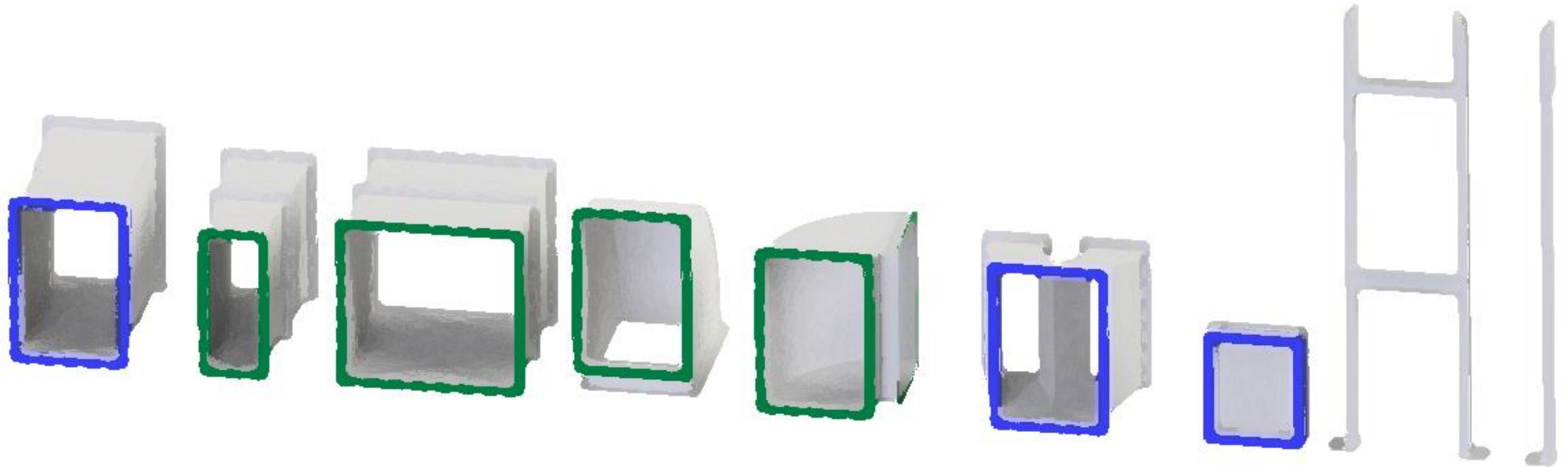
For Each refDoc As Inventor.ApprenticeServerDocument In Dok.AllReferences
    If System.IO.Path.GetFileNameWithoutExtension(refDoc.FullName) = "" Then
        Vorschlag = System.IO.Path.GetFileNameWithoutExtension(refDoc.FullName)
    Else
        Vorschlag = ReplaceName & "-" & System.IO.Path.GetFileNameWithoutExtension(refDoc.FullName)
    End If

    If Zeitstempel = True Then
        New_Name = NeuerPfad & "\" & Vorschlag & "-" & Zeit & System.DateTime.Now.ToString("yyMMddHHmmss")
    Else
        New_Name = NeuerPfad & "-" & Vorschlag & "-" & Zeit & System.DateTime.Now.ToString("yyMMddHHmmss")
    End If

    fileSave.AddFileToSave(refDoc, New_Name)
Next
```

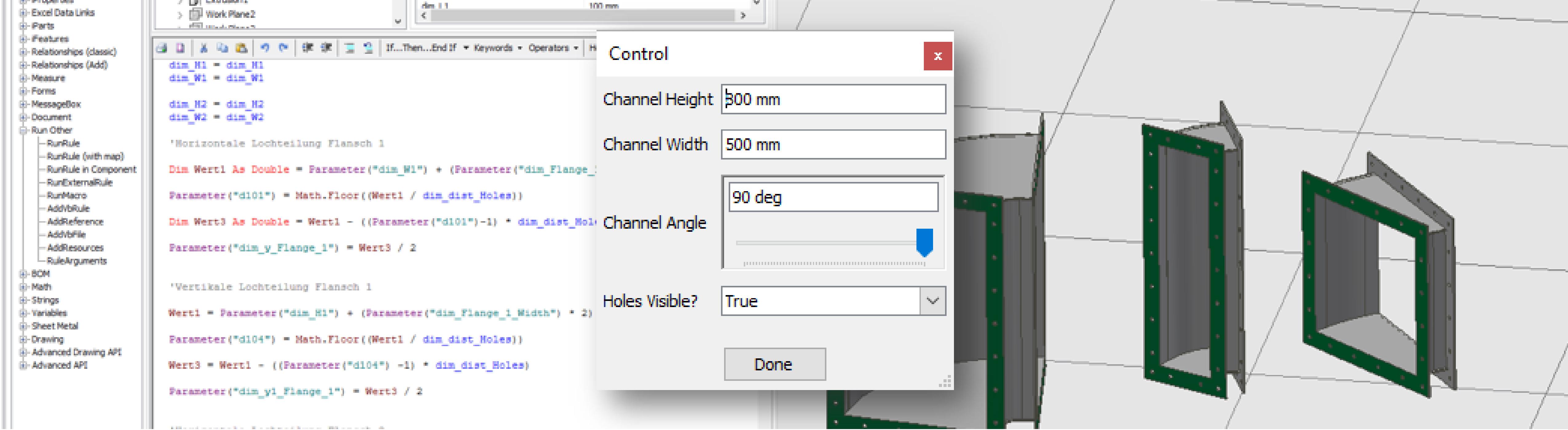
Umsetzung





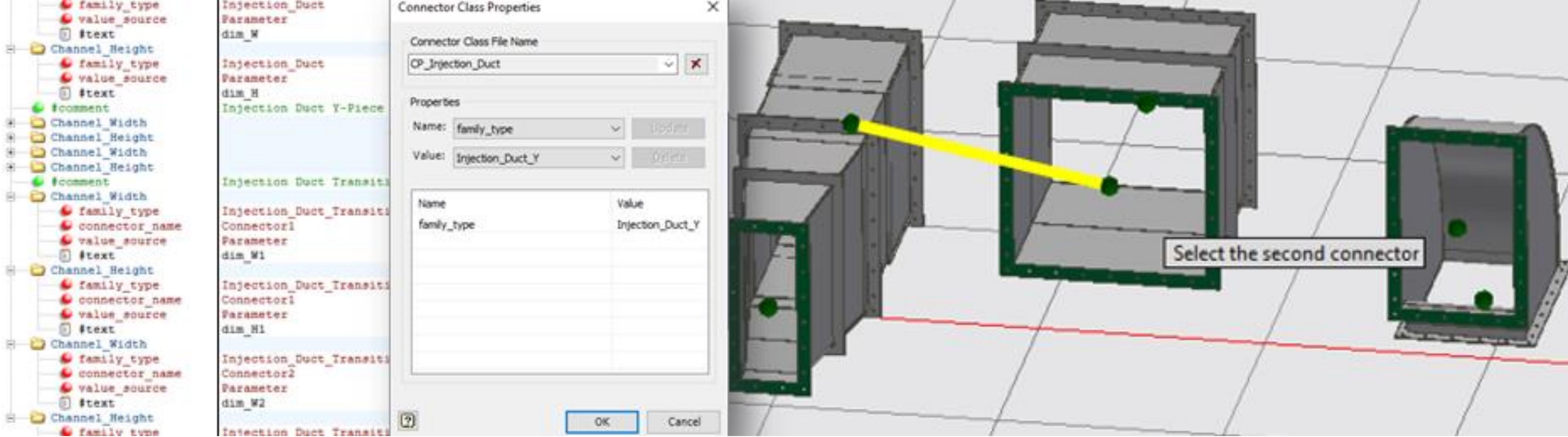
Erstellung der Assets

Erstellung der 9 erforderlichen Assets mit den entsprechenden Ausprägungen.



Programmierung aller Varianten in iLogic

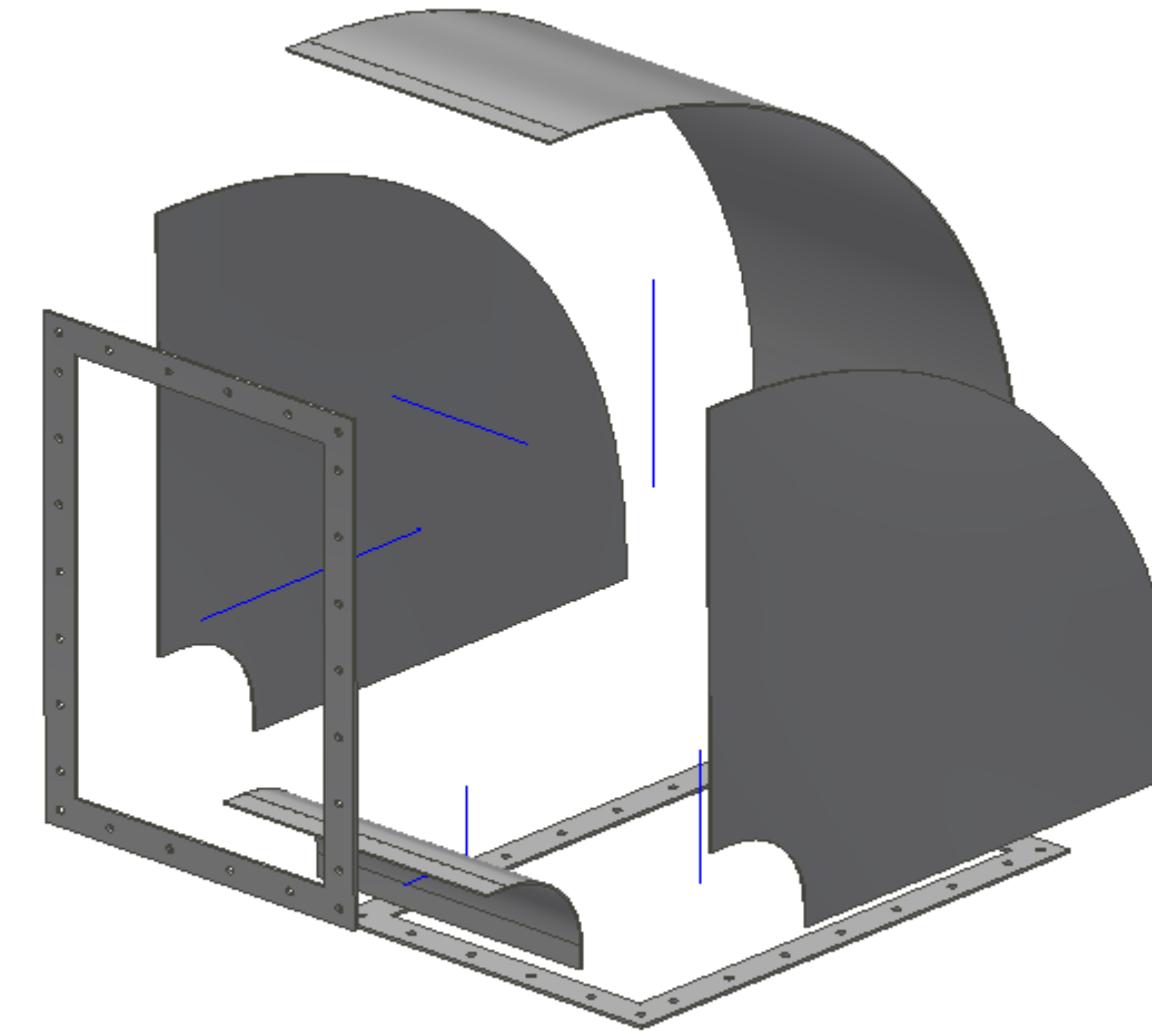
Erstellung der notwendigen Regeln und Parameter. Testen der Varianten durch Nutzung von Forms.



Definition der Connectoren

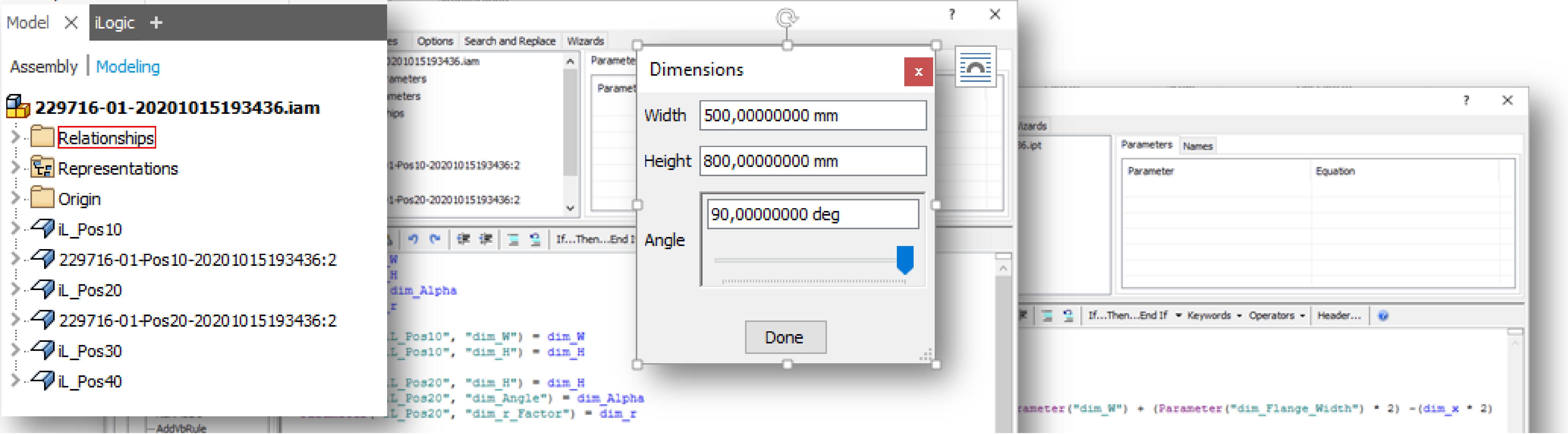
Erstellen der Connectorclass Datei.

Anschließend definieren der Anschlüsse vom Asset mit den entsprechenden Connectoren.



Erstellung der Fertigungsmodelle

Die Fertigungsmodelle werden so erstellt, wie es für die Fertigungsprozesse erforderlich ist und es in Inventor normalerweise getan wird. (Bleche mit dem Blechmodul etc.)



Programmierung der Fertigungsmodelle

Die Fertigungsmodelle müssen etwas anders programmiert werden, als die Assets, da hier eine Baugruppe die Steuerung für ihre Einzelzeile übernimmt. (Weiterleitung von Parametern und die entsprechende Reaktion darauf)

Part Number:

229716-01xxx

Wie findet das Asset ihr Fertigungsmodell?

Ganz einfach, im Asset wird eine Info hinterlegt (Hier bei uns als Part-No. in den iProperties), mit deren Hilfe das Fertigungsmodell im Vault gefunden werden kann.

Das xxx dient der Identifikation, das es sich wirklich um ein Asset mit Fertigungsmodell handelt.

33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

Create

Create Design Part Transfer Parameter

Timestamp Cancel

Overweise

searchModule Transfer_Parameter

condi' 3 Verweise

Dim sf

Dim St

For Each Element As Parameter In SourceDocument.ComponentDefinition.Parameters

If Element.IsKey Then

Try

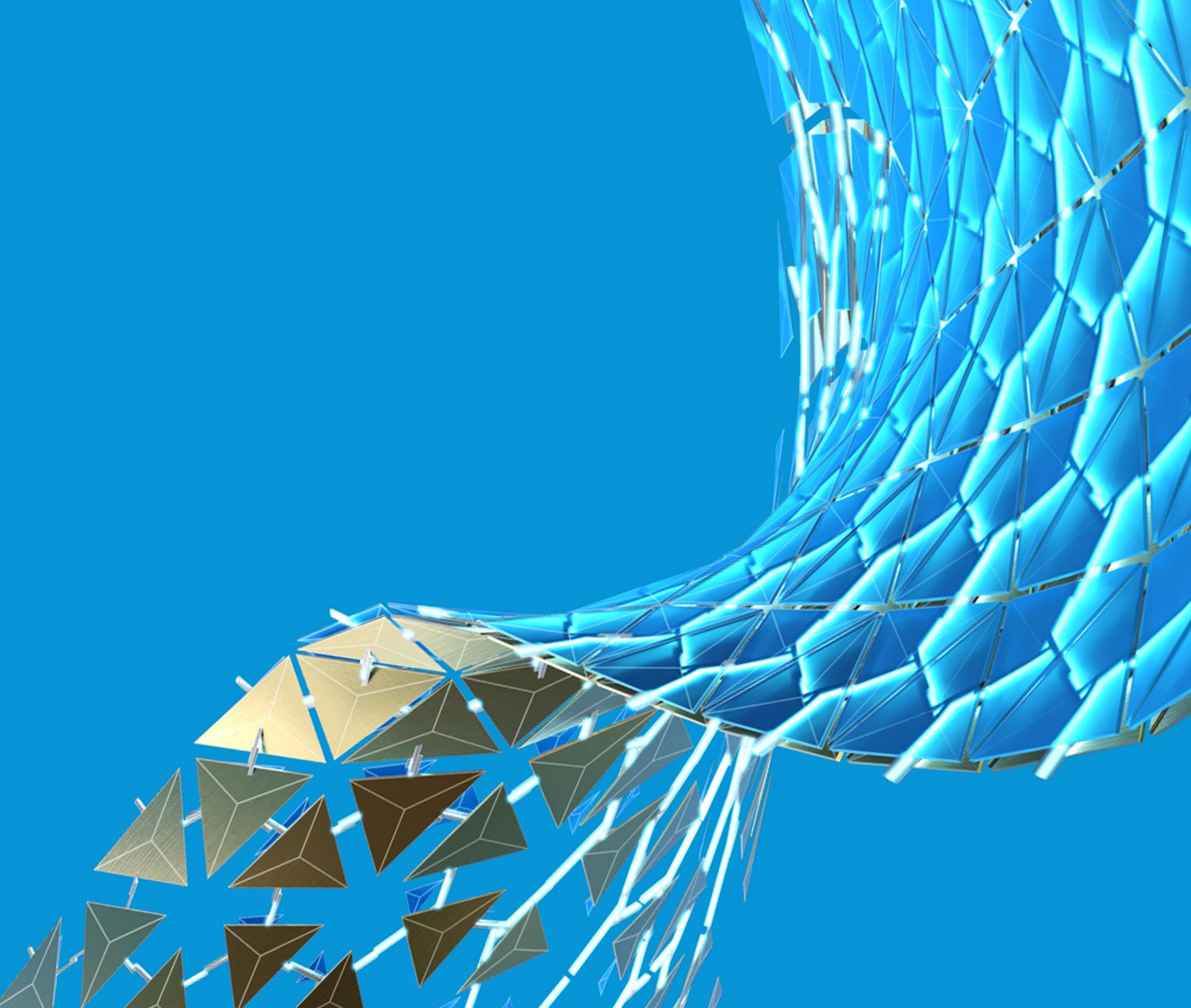
TargetDocument.Componentdefinition.Parameters.Item(Element.Name).Value = Element.Value

```
Module Transfer_Parameter
    Public Sub TransferParameter(SourceDocument As Inventor.Document, TargetDocument As Inventor.Document)
        Dim sf
        Dim St
        For Each Element As Parameter In SourceDocument.ComponentDefinition.Parameters
            If Element.IsKey Then
                Try
                    TargetDocument.Componentdefinition.Parameters.Item(Element.Name).Value = Element.Value
                End Try
            End If
        Next Element
    End Sub
End Module
```

Verbindung von Asset zum Fertigungsmodell

Per .net wurde dann noch die Verbindung vom Asset zum Fertigungsmodell erzeugt.

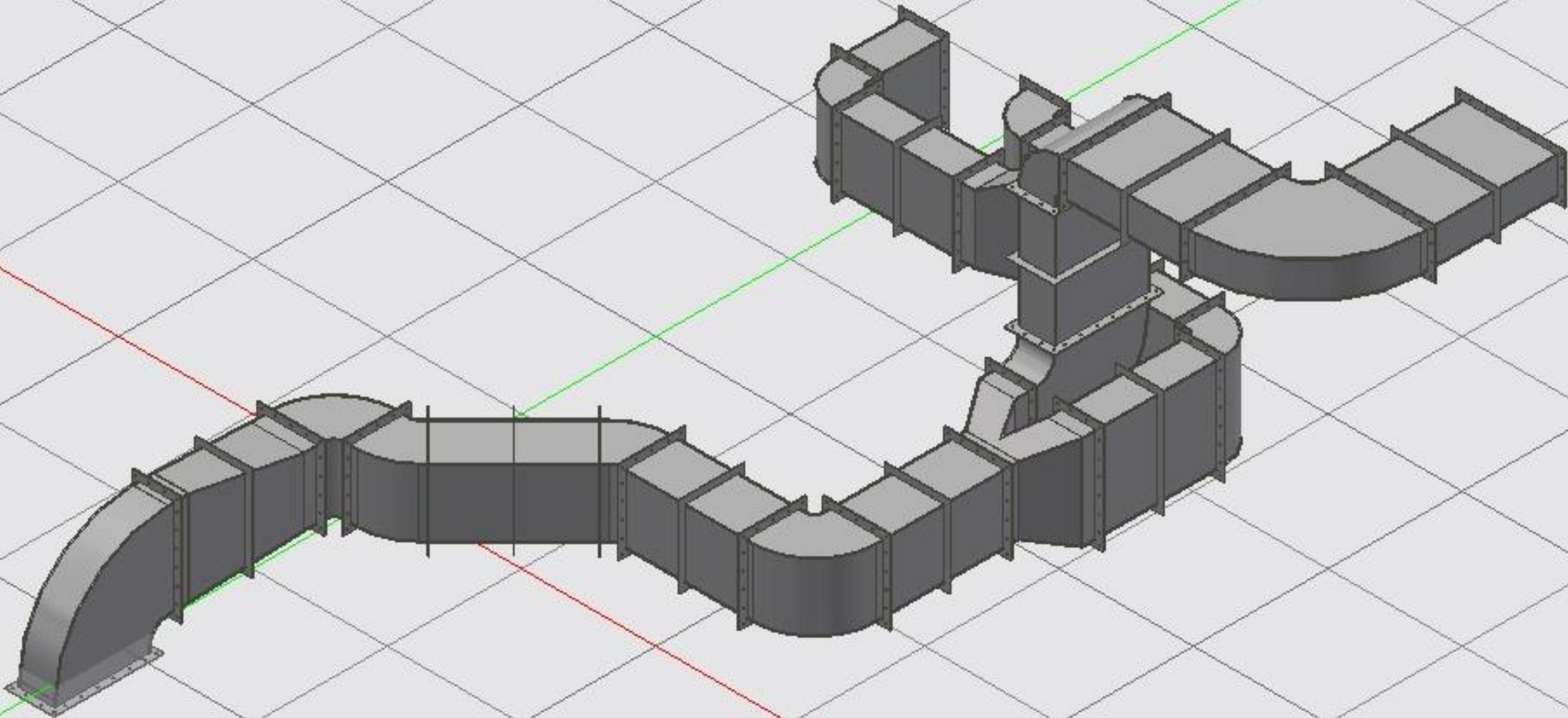
Ergebnis



Wie sieht das Ergebnis aus?

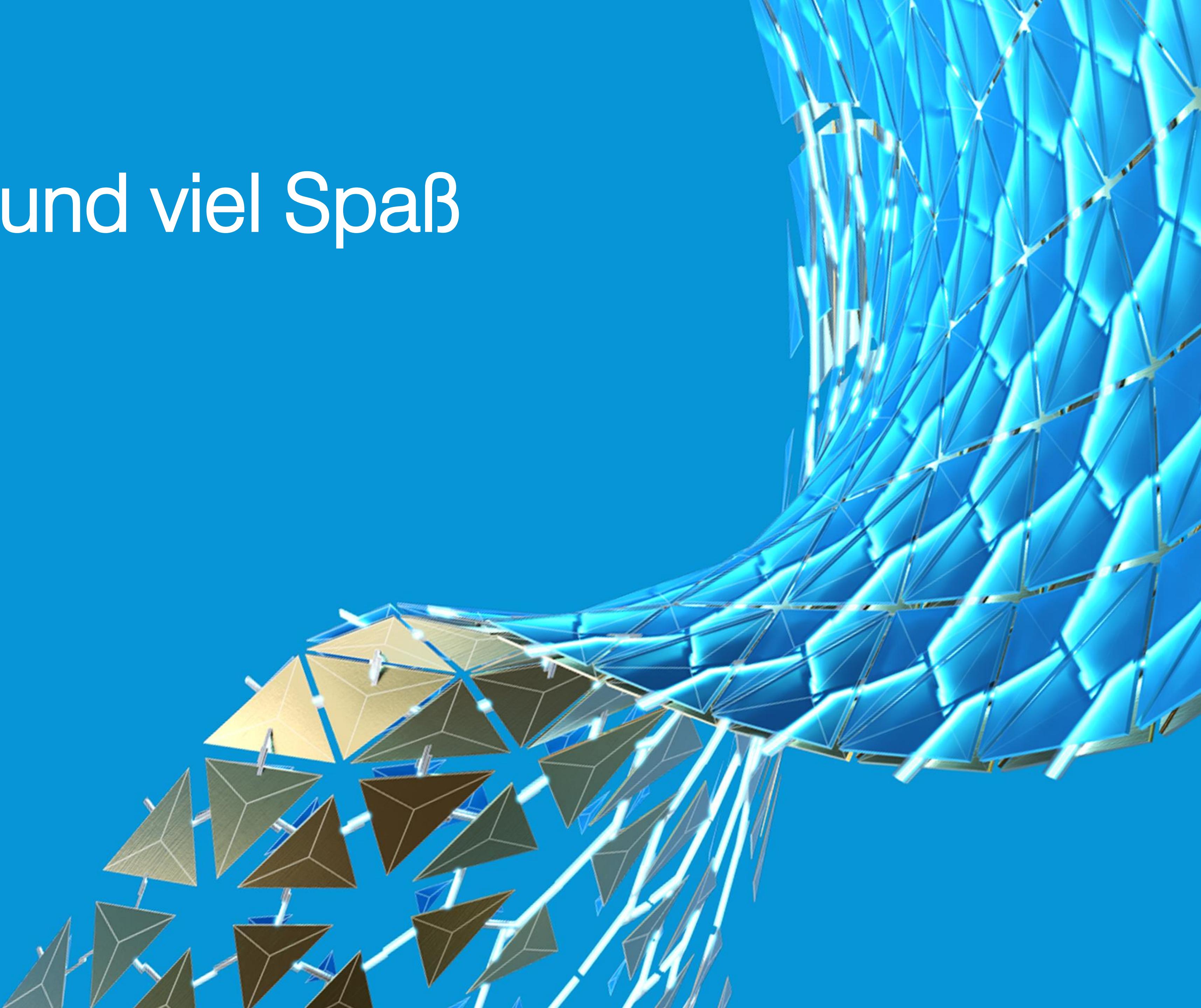
<https://youtu.be/KFefT2G5l5I>

Hier ein kurzes fiktives Beispiel einer Planung mit den Komponenten und dem anschließenden Ableiten in Modelle mit Fertigungsgenauigkeit.



Mindestens 50% Zeit gespart

Dankeschön und viel Spaß





Autodesk und das Autodesk-Logo sind Marken oder eingetragene Marken von Autodesk, Inc. und/oder ihren Tochtergesellschaften bzw. verbundenen Unternehmen in den USA und/oder anderen Ländern. Alle anderen Marken, Produktnamen und Kennzeichen gehören ihren jeweiligen Inhabern. Autodesk behält sich vor, Produkt- und Service-Angebote sowie Spezifikationen und Preise jederzeit ohne Vorankündigung zu ändern. Alle Angaben ohne Gewähr.

© 2020 Autodesk. Alle Rechte vorbehalten.

