BIM for Resilience: Automating Designs to Retrofit Informal Housing

Nicolas Ortiz Abello
Build Change

Allie Young
Build Change

Andres Felipe Robles
Build Change

Sebastian Moreno
Build Change

Learning Objectives

- Identify bottlenecks in existing workflows of retrofit projects.
- Learn about the key components of structural retrofitting techniques and strategies to automate them.
- Learn how to design the framework to support a BIM-automated tool, from data collection to drawing production.
- Understand the potential of technology for disaster risk reduction initiatives.

Description

In this class we will explore how technology - especially BIM and Automation - can improve disaster risk reduction. Since the early developments of new technology for design and construction, we have seen tremendous improvements that range from 3D visualization to cost and time efficiency in construction. These developments changed the way we design, analyze and build projects, leading to quality improvement, timely deliveries and a better performance in the AEC industry. Now think about climate change and Global Challenges such as safe housing, affecting millions around the globe. What if technology could mirror all the benefits from the AEC industry and at the same time help population in vulnerable conditions? This Industry Talk will show you how BIM and Automation made possible to deploy a nationwide project to strengthen 600,000 houses that are vulnerable to earthquakes.
Speakers

Nicolas Ortiz Abello
New Frontier Technologies Engineer
nicolas@buildchange.org

Allie Young
BIM Technologies Specialist
allie@buildchange.org

Andres Felipe Robles
BIM Specialist Engineer
andres@buildchange.org

Sebastian Moreno
BIM Specialist Architect
sebastian@buildchange.org
Introduction

By 2030 more than 3 billion people will be living in substandard housing conditions. This means that over a third of global population won’t have access to safe housing. Also, climate change and rapid urbanization are contributing to the increased levels of risk in developing countries. One of the main causes of substandard housing is informal construction, which takes place when low-income families move from rural to urban areas and build their homes without technical guidelines. This results in massive neighborhoods filled with poorly built houses that lack structural components necessary to withstand natural events like earthquakes and windstorms.

We talk about risk as the combination of vulnerability and hazard. Vulnerability refers to the level of exposure, quality of infrastructure, and how well prepared is a city or a country in terms of safety. On the other side, hazards are attributed to natural events and uncontrollable variables, like earthquakes and windstorms. In other words, vulnerability is a variable controlled by humans and hazard is not, and the overlapping of these two gives a measure of the risk status. Also, the two sides of the risk level can vary depending on location and context. For instance, Japan has a huge seismic hazard as it is located in an active seismic zone. However, Japanese building codes and regulations are particularly strict and their infrastructure is well suited to withstand this natural events. In total, Japan has a low risk towards earthquakes. Now, if we compare this scenario with a Latin-American country like Colombia, the situation changes. Similar to Japan, Colombia is an active seismic zone and it’s located in the intersection of 3 tectonic plates. On the other hand, Colombia has experienced an uncontrolled urban growth which has led to informal settlements and the rise of buildings that fail safety standards. Overall, this situation creates a big seismic risk that could affect 80% of the total urban population, because the larger concentration of buildings coincides with the higher seismic hazard zones.
This Industry Talk is the result of 6 years of work focused on preventing deaths from earthquakes in Colombia. In total, Build Change has been working for 15 years in 14 countries saving lives from earthquakes and windstorms. By working together with the ministry of housing, government agencies, and building authorities, Build Change has been an influential leader in the efforts to solve the substandard housing crisis through innovation and technology. Build Change’s approach for addressing unsafe housing is retrofitting, which essentially is strengthening an existing building that lacks structural integrity. This methodology is the most sustainable and cost efficient because it reduces the social impact of relocation and cuts down costs up to a third of reconstruction. With that in mind, the goal of this lecture is to share how technology and innovation scale up the initiatives to make a country safer and less vulnerable to earthquakes.

National House Improvement Program and Scalability Challenges

How to improve living conditions of 2 million people?

In response to an imminent seismic risk and to improve lives of over 2 million people, Colombia’s government envisioned a nationwide program to take place on a 4-year timeframe. The program “Casa Digna, Vida Digna” aims to improve 600,000 houses from an architectural and structural standpoint. Build Change was appointed as the technical consultant for the Ministry of Housing to design a framework that allows scalability and efficiency of the program. Our support to the national program focused on two areas: Structural Engineering and Process Innovation. First, we had to come up with a structural solution to make poorly-built houses safer, keeping in mind cost efficiency and local construction materials available. Once the engineering was set, we had to solve different challenges involving the bigger picture such as working in spread-out remote locations, lack of qualified labor, and more importantly, the magnitude of the program. In other
words, the engineering was not enough to implement “Casa Digna, Vida Digna”; we had to think of new workflows to ensure the program’s success.

**Traditional Workflow of Retrofit Projects**

Traditionally, the process of structurally retrofitting a building requires a sequence of steps that can be summarized into 4 stages: **Data Collection, Data Processing, Document Production** and **Construction**. The workflow starts once a house has been identified as a potential beneficiary and all the requirements for subsidy are met. Once the subsidy procedures are completed, trained staff must visit the house to do an initial assessment and collect data. Then, this data is taken into the office where analysis is performed to define which structural solution should be implemented. Once the retrofit design is defined, engineers and architects assemble a set of construction drawings that must be approved by the building authority to start with the construction activities.

![Traditional Retrofitting Workflow, from field data collection to construction of the retrofit solution.](image)

1. **Data Collection:**
   This part of the process takes place in the field, at the house intended to retrofit. It’s done by trained staff, either civil engineers, architects or professionals instructed in earthquake resistant construction. In this stage two types of data are collected: A geometrical survey of the house, which records all the measurements of walls, windows, doors and floors in a hand sketch. Then, the second data type is homeowner information together with seismic site parameters to calculate the vulnerability status.
2. **Data Processing:**
   This stage includes a vulnerability assessment, followed by a structural analysis which will establish the retrofitting techniques required. They breakdown as follows:

- **Vulnerability Assessment:**
  The factors that determine the degree of vulnerability of a house include site hazards (such as landslides or floods), seismic parameters intrinsic to the area and the structure configuration.

- **Structural Analysis:**
  Informal housing lacks a proper design and often are built without fundamental structural elements. Therefore, structural engineers must analyze the structure configuration and the materials used to calculate a seismic demand for the building.

- **Retrofit Proposal:**
  As a result of combining the vulnerability assessment and the structural analysis of the building, engineers come up with a retrofit proposal. This includes a set of structural elements such as columns, beams, ties, dowels and other solutions to ensure the building won’t collapse during an earthquake.

3. **Document Production**
   The third step of the process is crucial because it wraps everything done until now in a concise package of information known as the Construction Package. In other words, this document portrays a timeline for the house intervention because it shows the initial conditions of the structure, the results of the analysis and what’s needed to make it safer. In detail, this construction package includes:
• Existing plans of the building, including architectural and structural elements
• Results from the structural analysis that determine the intervention’s scope
• Retrofit plans of the building, specifying materials and elements for the intervention
• Construction details of the retrofit elements
• Cost estimate or Bill of Quantities (BOQ)

4. Construction
Once the construction package is approved by the local building authority, the project has green light to begin construction works. The retrofit solution includes the construction of new elements such as columns, beams or jacketing, and also modification of existing walls, slabs and existing structural components.

Bottlenecks and Limitations for Scalability

The previous retrofitting workflow has been implemented by Build Change in 7 countries. However, it has never been used in a large-scale National Development. With this in mind, to effectively implement a nationwide program for retrofitting, this workflow had to be reassessed from a cost and time perspective. The image below shows a map of the workflow, focusing on processing time and information exchange.

A systems-based analysis of the traditional workflow revealed several bottlenecks that limited it’s use for a nationwide implementation. The stages that represented most of the inefficiency were Data Collection, Data Processing and Document Production. In the current workflow, information
was not integrated on a single platform and different data types implied extra work for conversion and processing. The most significant bottlenecks were identified below:

- Data collection implied using trained resources such as engineers or architects to visit houses to conduct the geometric survey and vulnerability assessment. A large-scale implementation would be extremely costly using this scheme and the scarcity of trained staff could slow down the program.

- The process of collecting geometric data of the house using a hand sketch was time consuming (it could take up to 3 hours) and produced inaccuracies down the line. For instance, during the design phase an engineer could find a missing measurement which forced the design process to stop until field staff confirms this missing value.

- An engineer had to convert the hand sketch into a Revit model that represents the existing conditions. This required interpretation skills from the engineer and often led to errors in the model. Depending on the complexity of the house, this process could take up to 2 days.

- The structural analysis was done using Excel spreadsheets that calculated shear and gravity loads. General behavior of the structure was verified through static linear analysis, checking stress concentrations and different failure modes. This engineering methodology is not suitable for a large-scale implementation because it focuses efforts in a case-by-case analysis.

- Once the analysis defined the retrofit intervention, engineers proceed to add new structural elements to the model. This process involves a lot of Revit adjustments, dealing with phasing, element parameters and graphics. The production of the Construction Package for one house could take up to 9 days, taking into account rework caused by data inaccuracies and retake of measurements in field.

- As a whole, there wasn’t a platform that integrates data from all stages to manage the project. Progress was tracked using an online spreadsheet but there was no way of incorporating all data types into one platform to have a global understanding of progress, delays and performance.

In summary, the existing methodology for retrofitting houses was not optimal for a large-scale deployment. The workflow heavily relied on qualified labor, which increased the cost considering the number of houses that would be retrofitted simultaneously. Also, the long processing times in design and document production phases, made the overall operation too expensive for scaling it up. For instance, a typical two-story house would need a team of 2 trained professionals to assess and take measurements on site (usually civil engineers or architects), plus a structural engineer in the office performing analysis and retrofit design, and a drafter (could also be an engineer or architect) that puts together the set of plans, calculations and cost estimate in a construction package. This whole workflow normally takes from 5 to 9 working days on full schedule, and varies depending on the complexity of the house. Now, considering the magnitude of the National Program we are addressing and the number of houses to deal with, this workflow results non-viable in terms of costs and time.
Proposed Workflow for Large-Scale Implementation

The reassessment of the retrofitting workflow showed key aspects to improve in order to make the national home improvement subsidy program feasible from a technical perspective. Qualified labor reliance, time spent in designs and document production were targeted for a redesign. The goal was to make a smooth workflow that streamlined the production of code-compliant retrofit designs with a focus on cost and time efficiency.

The solution we found to improve our workflow is based on 3rd party apps integration and BIM-Automated tools. By setting our priority on the time spent in each process, we managed to overcome interoperability barriers between different software used. With this in mind, each bottleneck was addressed with a particular solution that, altogether, represented time savings of up to 78% in the workflow. A breakdown of the bottlenecks and their solution is summarized in the table below:

<table>
<thead>
<tr>
<th>Bottleneck</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliance of highly qualified labor to assess and collect house data</td>
<td>Use of 3rd party mobile apps, tablets and GIS data</td>
</tr>
<tr>
<td>Hand Sketch for geometric survey of the house</td>
<td>Dynamo Scripts for Automated Existing Model Generation</td>
</tr>
<tr>
<td>Manual modeling of the existing house in Revit</td>
<td>Prescriptive Structural Analysis and simplified rule checks</td>
</tr>
<tr>
<td>Traditional Structural Analysis using Excel spreadsheets</td>
<td>Dynamo Scripts for Automated Structural Checks, Retrofit</td>
</tr>
<tr>
<td>Manual adjustment of the existing Revit model to include retrofit elements</td>
<td>Modeling and Cost Estimate</td>
</tr>
<tr>
<td>Preparing the construction package including cost estimate</td>
<td>Web-Based Platform connecting field data with designs</td>
</tr>
<tr>
<td>Lack of an integration platform for management</td>
<td></td>
</tr>
</tbody>
</table>

Bottlenecks found in the traditional workflow and tech-based solutions to address each one.

For the most part, automation of BIM tools impacted the most in the overall performance of the workflow, creating the biggest savings in time and effort. On the other hand, use of 3rd party apps and the platform integration made possible to keep track of every house on all stages, and kept information organized throughout the process. Each one of these solutions will be discussed with more detail in the next chapter.
How we did it: Framework for BIM-Automated Tools

In a nutshell, we reduced the processing time per house by 78% while keeping the thoroughness of structural safety analysis and quality. The keys for this improvement were 3rd party app integration and automation of our BIM tools. By tweaking our existing workflows and adjusting the engineering methodology, we achieved technical and economic feasibility to implement the large-scale national program. The new workflow reduced the reliance on a structural engineer in the assessment and design phases, allowing non-specialized staff to produce code compliant structural designs automatically. In other words, we adjusted our engineering methods to allow their automation. This streamlined the design production without using costly resources (trained structural engineers) and focusing them at the end of the process for revision and approval.

Data Collection

In the traditional workflow, the information was registered using paper and predefined forms to gather data from the house and homeowners. This caused significant downtimes and delays down the line because of inaccuracies in measurements, missing information, human errors and inaccuracies. The way we tackled this issue was by integrating mobile apps that could be used in smartphones or tablets, even in an offline mode. We divided our data collection in 2 groups: geometric survey and general information survey. Having our data divided in two categories enabled us to use specific apps to deal with different types, simplifying the overall process.

Magic Plan

This mobile app allows you to draw house floorplans including walls, windows, doors, floors, and other existing elements with your fingertips. Using predefined templates, we created a standardized protocol to draw house floorplans including key elements for the design phase. Also, the app can be linked to a laser measure device to speed up the process. Once the survey is completed and internet connection is available, the plan is uploaded to the cloud where it can be downloaded from the office or any location.
Fulcrum

The second portion of the data collection happened in Fulcrum. This mobile app is widely used in construction, manufacturing, and logistics for data capture and creation of smart forms. We incorporated this app into our workflow to collect and process information based on location. The form builder allowed us to design mistake-proof surveys that guided the surveyor through the questionnaire with adaptive responses and skip logic which made the process simple and efficient. More importantly, Fulcrum allowed integration with GIS that included seismic and vulnerability parameters from official databases. In this way, detailed information that used to be processed in the office is now preloaded in a digital form and ready to be used in field.
The integration of these 3rd party apps into the system, improved data collection making it more accurate, faster and intuitive enough to be used by a non-technical surveyor. Similarly, by integrating GIS information from vulnerability and seismic hazard databases, the assessment is done by the app, removing the need for a structural engineer on site. This improvement was achieved by overlapping location-based data such as peak ground acceleration and soil type with house configuration data (wall lengths, location and inter-story height) to calculate vulnerability status. In this way, the data collection could now be performed by social mobilizers, students, volunteers or anyone with a few hours of training. With this we had a massive cost reduction, by allowing anyone with some instruction on how to use the apps to collect data. Just imagine training people remotely on how to take geometric surveys all around Colombia in a few minutes with an app and a smartphone. That's what scalability is all about.

Once the data is collected using Magic Plan and Fulcrum, both data packages are sent to a web based platform developed specifically for this program by Build Change. Essentially, this platform receives the plan made in Magic Plan and links it with the Fulcrum data using the homeowner name and ID. At this point, information from the house measurements, materials, configuration and vulnerability status are stored in an XML file.

**Existing Model Generation**

Right after data has been collected and stored in the digital cloud, the second step is to retrieve the information and, afterwards, model the existing conditions of the house. Traditionally this was done by looking at a hand sketch with measurements and annotations of the house and manually modeling it in Revit. Now, with the data collection digitalized and stored as an XML file, this process improved substantially. To automate this entire process, we designed visual programming scripts in Dynamo to get a Revit model as a final outcome. Here is how we did it.

**Importing the plan:**

The information of all the plans is stored inside Magicplan’s cloud. This cloud works in a business account called a workspace, which can store the information of the plans uploaded by several accounts that belong to said workspace. The way to access this information is by making a web request to Magicplan’s API, which will get us an equivalent response. This a standard client-server interaction.
However, this interaction is not possible with the OOTB nodes from dynamo, which is why we used an external package for making web requests, called Dynaweb.

We want to make the user experience as smooth as possible, which is why we are connecting them directly with the list of plan names, so that they can pick a plan, and based on that choice, the script can retrieve the file information of the plan. This means that we must make two HTTP requests to the server:

- Get the list of plans
- Get the files of a specific plan

To get these requests, we need some parameters, starting with credentials to access the API. The owner of the workspace has these, which are the API_SECRET_KEY and the CUSTOMER_ID.

<table>
<thead>
<tr>
<th>API key</th>
<th>Customer ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>API_SECRET_KEY</td>
<td>CUSTOMER_ID</td>
</tr>
</tbody>
</table>

Magicplan's workspace owner credentials for the API key

Once we define the main actions that we need to do, we can establish the order of those actions to outline the workflow of the Import Script.

Once we define the main actions that we need to do, we can establish the order of those actions to outline the workflow of the Import Script.

You can see how the web requests look like using Dynaweb:
Notice that response we get is a string in a JSON format, which we need to parse into a Dynamo dictionary to read the information easily, hence the blue conversion icon on the workflow.

We are also using a UI (User Interface) provided by the Data-shapes package to get the input we need from the user from the list of plan names.
Finally, by executing all the tasks, we got to our final import script:

![Diagram of the import script]

Processing the plan:

After the file is saved locally, we need to read it and process the information to create elements. This is possible by first understanding the structure of the xml file which has a nested structure enclosed by the tags: “plan”, “floor”, “floorRoom”, ”point”, “window”, “door”, all of which contain information (attributes) that we need for the definition of the elements or for establishing parameters. The hierarchy of the tags works as follows:
A Python script deconstructed the list of values in the XML file and arranged the values needed by Dynamo to model wall, windows, doors, floors and slabs. There's different values for each element, so they must be organized per type as follows:

- **Walls**
  - X and Y coordinates for start and end points
  - Wall height
  - Wall type (thickness and material)
  - Windows and doors
  - Overrides

- **Windows and Doors**
  - X and Y coordinates of the insertion point
  - Width, Height, Sill Height
  - Window/Door type
  - Overrides

- **Slabs and Floors**
  - X and Y coordinates of perimeter points
  - Elevation from level zero
  - Floor type (thickness and material)
  - Overrides
This process implies that we need to re-organize the information, filter it and categorize it to adjust to the information that we ultimately want to have for the creation of elements.

To this end, we used Python’s OOP (Object Oriented Programming), which allows us to create classes and shape them with customized attributes that would be extracted at specific points of the xml file. In addition to this, OOP also allows us to create methods for those classes, which are functions applied to the objects. This is of great use and benefit because we can use the multiple list of attributes of a determined class in a much more organized way, without having to use nested lists, which can get messy with a long list of elements, each of them holding a long list of attributes, some of them possibly nested also. Here you can see the definition of those classes and the methods later used to create the elements:

```python
class Window_Door:
    def __init__(self, x, y, w, h, h_p, dw_type, opening_unconected, hlev):
        self.x = x
        self.y = y
        self.h = h
        self.hlev = hlev
        self.w = w
        self.h_p = h_p
        self.dw_type = dw_type
        self.opening = opening_unconected
```

```python
class Wall_Line:
    def __init__(self, g_line, w_h, w_type, ds_ws=[], overrides={}):
        self.line = g_line
        self.height = w_h
        self.wtype = w_type
        self.windows_doors = ds_ws
        self.overrides = overrides
```

```python
class Floor_Curves:
    def __init__(self, polcurve, floortype, overrides = {}, isroof = False, projparams = {}):
        self.polcurve = polcurve
        self.floortype = floortype
        self.overrides = overrides
        self.isroof = isroof
        self.projparams = projparams
```

Additionally, the classes defined have management methods, such as a comparison method for the Wall Line class, which evaluates if the spatial information of two objects intersect and modifies the walls accordingly. However, the most important method of the classes is the method to create the elements, which takes the information of the attributes of the object and creates the Revit element.
Class methods to evaluate the information of the objects and create Revit elements

The final structure of the classes is shown below:

```python
class Wall_Line:
    def __init__(self, g_line,w_h,w_type,ds_ws= [],overrides={}):
        ...
    def createElements(self,level):
        ...
    def get_all_points(self):
        ...
    def divide_by_openings(self,minw_width):
        ...
    def compareByBox(self,w_line):
        ...
    def addDoor(self,w_door):
        ...
    def checkAngle(self):
        ...
    def correct_points(self,x_snaps,y_snaps):
        ...

class Window_Door:
    def __init__(self,x,y,h,w,h_p,dw_type,opening_unconnected,hlev):
        ...
    def createElements(self,level,host):
        ...
    def correct_points(self,x_snaps,y_snaps):
        ...
    def compare(self,wind_door):
        ...
    def get_all_points(self):
        ...

class Floor_Curves:
    def __init__(self,policurve,floortype,overrides = {},isroof = False, projparams = {}):
        ...
    def createElements(self,level):
        ...
```

Final Structure of the classes
At this point, we’ve modeled the existing conditions of the house with exact measurements taken in field using Magic Plan, and a vulnerability status that resulted from Fulcrum data processed with seismic and soil parameters. The time spent in the existing house modelling went from 1 or 2 days to 10 minutes. Now, the house is ready to begin the structural analysis to determine the best retrofit solution.

Adjusting the Engineering Design Methodology

To understand the engineering behind retrofitting, we must discuss the types of houses it aims to strengthen. In Colombia, over 60% of residential buildings are 1 to 3 story houses made with horizontally perforated clay blocks. The predominant building typology found in informal settlements is unreinforced masonry, which is highly restricted by the national building code and is not allowed for multistory houses. In this types of buildings, the structural elements that bear the gravity and seismic loads are masonry walls, which are often disconnected to diaphragms or adjacent walls.
Traditional house in an informal neighborhood in Bogota, Colombia, with mixed types of blocks and lacking a defined structural system.

Essentially, structural retrofitting ensures that the building’s resistance is greater than the seismic demand. The factors affecting these two sides of the equation are structural configuration, materials used, quality of construction and foundations. In terms of failure modes, the most common are in-plane and out of the plane failures due to shear stress concentrations. Slender walls and lack of connections are common causes for this failure modes. Tension and compression failures are also found when the width or effective area of blocks is insufficient for gravity loads.

In-plane failure modes common in unreinforced masonry walls
To improve the seismic performance of informal masonry houses two things can be done related to the following resistance vs demand inequality.

\[ \text{Resistance} \geq \text{Seismic Demand} \]

First, on the resistance side, the only structural element bearing the seismic and gravity loads are masonry walls. Considering that walls were not built under reinforced masonry system considerations, the way of measuring the resistance as a whole is through the Wall Area Percentage (WAP). This represents the cross section area of blocks that can transfer loads to the foundation system. Adjusting the Wall Area Percentage by factors based on materials and width of the block, the result gives us an accurate estimation of the resistance of the house.

\[ WAP = \frac{\text{Cross section Wall Area}}{\text{Area of the House}} \]
The WAP of the existing house is insufficient in almost every case of informal unreinforced masonry buildings. The strategy to increase this value and add resistance to the house is through either adding more walls, closing openings or plastering (with simple or mesh plaster) existing walls. The result of these strategies increases the resistance of the house and improves the load distribution. However, increasing resistance through WAP sometimes might not reach values greater than seismic demand. In this cases, engineers need to reduce the demand, to make it smaller than the resistance obtained in the previous step.

On the other side of the inequality, Demand can be substantially reduced by modifying the structural configuration. In contrast with the previous strategy of increasing resistance through new walls and plasters, structural system adjustment can reduce seismic demand because it changes how the building behaves in an earthquake. Keeping in mind that more than 90% of the houses in this case study are made of unreinforced masonry, the resulting demand tends to be extremely high due to lack of ductility. Therefore, the best way to reduce seismic demand is to confine the existing walls using tie columns and beams. Also, by connecting free wall edges and creating closed loops of walls, the overall behavior of the building improves notably.
Conversion from an unreinforced masonry system into a confined system by adding tie concrete elements

Using both strategies, increasing resistance and reducing seismic demand produce the final scope of work for the retrofit solution. The retrofit solution for a typical 2 story informal house would include the following major retrofitting interventions:

- Add 3 cm plaster in perimeter walls, both sides
- Add dowels to connect walls to the foundation or inter story slab
- Confine walls using concrete tie columns and ring beams
- Extend upper portion of the openings to match the ring beam
- Add a new confined wall attached to walls that are unsupported
- Complete short walls to match bottom of ring beam and connect them

This design methodology was performed using customized spreadsheets that included seismic parameters from the building code and all the formulas to calculate stress, deformations, and load combinations. Despite the thoroughness and standardization of these spreadsheets, they were not efficient because they operate on a case-by-case basis. As discussed in the previous chapter, one major bottleneck was the design phase which relied on a structural engineering and spreadsheets, taking up to a week to be completed. Considering the substantial gains in efficiency from the data collection, especially the removal of field verification and rework due to inaccuracies, we targeted the structural design phase for a major process improvement.
The first step towards improving the current design methodology was to look for patterns in structural deficiencies and types of solutions coming from traditional case-by-case designs. After 6 years of experience working in Colombia and a series of lab testing in partnership with local universities, Build Change found that most of the houses could be categorized, and all houses in each category shared the same retrofit solutions. This laid the foundations for what we call a Prescriptive Design Methodology which aims to produce structural designs with the same quality and thoroughness as the traditional methodology, but in a fraction of the time. The Prescriptive Design Methodology is based on checking a series of structural rule checks to houses previously categorized, and then applying a common retrofit solution. Looking at the big picture, this new methodology fits smoothly in the whole workflow because it removes the engineer from the equation and leaves him or her at the end of the process to approve the results of the Prescriptive Design.

Once the patterns in deficiencies were identified, the scope of the Prescriptive Methodology had to be limited to common cases found in field. This was essentially the purpose of improving the current design method, and the goal was to develop a simple yet thorough series of rules that produce designs for the most houses possible. With this consideration, the first exceptions of the new design methodology were houses with concrete frames (very rare to find in informal neighborhoods), masonry houses of 4 or more stories, and multistory houses with cantilevers. These typologies had to be processed using the case-by-case methodology conducted by a structural engineer. After the scope definition, we began the process of assigning common retrofitting solutions through concise rule checks that went over the most critic aspects of structural deficiencies. As the name suggests, the prescriptive design enforces the use of particular interventions to solve a deficiency that could have several fixes. However, this new methodology prioritizes on commonly used interventions known by local builders. An example of the first category of houses and the prescriptive solution is summarized in the table below:
Summary of the Category addressing one story houses made with unreinforced masonry, which accounts for about 92% of low income housing in Colombia’s largest cities.

The new Prescriptive Design Methodology broke down complex engineering calculations into discrete rule checks that could be scripted and automated. This set of rule checks only required geometrical inputs and material properties to obtain a retrofit solution. In comparison to the previous “case-by-case” methodology, structural engineers conducted the calculations with spreadsheets and had to come up with a mix of interventions. Now, the Prescriptive set of rule checks prioritize on items that ensure a proper seismic performance and produce code-compliant buildings.

Automated Retrofit Designs and Document Production

So far in the newly developed workflow for the “Casa Digna, Vida Digna” National Program, we have discussed improvements in the Data Collection and Design Methodology phases. The third step towards creating an efficient and scalable workflow is automation for Retrofit Designs and Document Production. In this phase, the resulting model of the data collection is modified with the product of a prescriptive design methodology, creating the Retrofit Solution Model.

Traditionally this phase was done by the structural engineer once the retrofit intervention was defined. Then, the new elements such as tie columns, beams or plasters were manually placed in the Existing Revit model. Each retrofit intervention must be supported with the corresponding construction details in 2D. These come from an extensive library produced by Build Change and
are stored as drafting views in the customized Revit template. So once the new elements are placed, the engineer must manually add the right set of details to ensure a proper construction, following the quality assurance guidelines. Last but not least, in the traditional workflow the engineer had to calculate the Bill of Quantities (BOQ) of the intervention, by filling quantities in a spreadsheet that has preloaded costs of each retrofit component. The result after several days of work was a PDF package that we call Construction Package. This package includes:

- Set of Existing drawings including plans, sections and elevations including architectural and structural components.
- Set of Retrofit drawings, showing how the new retrofit elements fit in the existing structure.
- Set of Construction Details corresponding to each component of the intervention, ensuring that the retrofit works will meet earthquake resistant standards.
- Bill of Quantities, estimating a total cost of the retrofit including materials, labor costs and equipment.

The process innovation that made this phase of the workflow scalable and efficient was automation of our BIM models. Similar to the automatic modeling of the existing building, we developed Dynamo scripts to automate the rule checks defined by the Prescriptive Design Methodology. This new methodology was thought in a way that each rule check was independent and analyzed specific portions of the structural system. This features enabled us to program each rule check in Dynamo. With this in mind, the structural design in the new workflow will be performed in discrete steps (rule checks) and each will be linked to a cost (including labor, materials and equipment) and also associated with a particular 2D construction detail. This is the essence of the automated retrofit design production because it breaks down the process into steps, that then compose the retrofit intervention with cost and construction documents for each one of them. The proposed retrofit design breakdown structure is shown below:
Each rule check is verifying particular geometrical and configuration values in the existing model and compares them with the predefined design parameters. The outcome of a rule check might be compliant (the existing model doesn't present structural deficiencies), or non-compliant and new retrofit elements must be placed. Now, there are rule checks that sequentially check different aspects of the building and will throw a particular solution depending on each case. This is the case of the Front Wall Rule Check, that verifies if the front façade wall is rigid enough to withstand stress caused by torsion effects. This is checked in 3 steps, each with different considerations on how to fix possible deficiencies as shown below:

<table>
<thead>
<tr>
<th>Category</th>
<th>Prescriptive Design</th>
<th>New Elements</th>
<th>Outputs for Construction Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Check 1</td>
<td>Retrofit Outcome - A</td>
<td>BOQ - A</td>
<td>Detail - A</td>
</tr>
<tr>
<td>Rule Check 2</td>
<td>Retrofit Outcome - B</td>
<td>BOQ - B</td>
<td>Detail - B</td>
</tr>
<tr>
<td>Rule Check 3</td>
<td>Retrofit Outcome - C</td>
<td>BOQ - C</td>
<td>Detail - C</td>
</tr>
</tbody>
</table>

Retrofit Design Breakdown Structure: Each rule check has different outcomes which are linked to cost estimates and construction details through codes.
Simplified flow diagram of the front façade rule check and the sub steps that define a retrofit solution depending on the existing house configuration.

The front façade rule check has more than one possible outcome. This could be programmed into Dynamo using “Pass-through” Nodes from the Archi-lab package and a bunch of “If” nodes, replicating the flow diagram logic. However, there are other rule checks that look for specific conditions and if they are non-compliant according to the design parameters, the retrofit solution has only one outcome. This is the case of the Door Confinement Rule Check, that makes sure there are no doors without confinement tie columns at each side of the opening. The logic behind this rule check is explained below:

1. Find base line geometry of the door.
2. Extend these lines by the half of the hosting wall width to ensure that the column’s and door’s geometry won’t crash.
3. Get the start and end points of each of these lines to get the starting base points of the column’s axis, then project these points upwards until they reach the hosting walls height to get the top end points of the column’s axis.
4. Create a line between each pair of base and top points to get the columns axis.
5. Place tie columns on each column’s axis created earlier and set the cross section according to the design’s standards preloaded.
The following example shows a simple house with a Free Wall Edge condition. Let’s see how the Dynamo Script is structured and the different strategies used to produce a retrofit design automatically through visual programming.

Sample house to see the functioning of the Door Confinement Script and how it links BOQ, Details to deliver an automated retrofit design. The doors colored in green are the ones that need to be confined.

Once the logic of the rule check is mapped out, the script needs to follow each step and perform the calculations required for the prescriptive design. In addition to the 5 steps described above, the script also needs to link the corresponding code for the cost estimate and the construction details. Let’s see how the general structure of the script works and how the groups perform each task of the rule check:

General overview of the Free Wall Edge Rule Check script, with the most important groups marked from 1 to 6.
Group 1 gathers all the doors and walls in the Revit model and filters them by height (we don’t want to take into consideration walls shorter than 1.2 m). Then, Group 2 deconstructs the doors by geometry and then by start and end point. Let’s expand on Group 2 and see how it works. To get the tie column in its right place we first need to get the starting base points, we do this getting the base line of the door and then extending it in each one of its ends, we must do this to ensure that the column will not interfere with the door opening, but will be placed at each side creating a correct confinement.

Strategy to find the columns starting base point using door geometry and extending them by the half of the width of the door's hosting wall.

Going back to the general overview of the scripts, we have identified the starting points where a tie column must be placed. Then, Group 3 of the script takes these points and projects them on the positive Z direction until it reaches the top height of the existing walls. The second point represents the end point of the tie column. Following the projection, a node creates a line by start and end point and the base geometry for the column is done. Next, the group 4 creates the actual columns using the lines, type of column and level as inputs.

The new tie columns have been created along the axis lines.
As shown in the figure above, once the new columns are created, the Phase Created parameter should be set to “Retrofit” so it will display properly in the set of drawings. Once this script is run, several tie columns have been created under the Retrofit phase. The picture below shows the result of the script on a simple house.

![Finished product of the Door Confinement rule check script.](image)

Now that the new elements have been placed, the cost estimation and set of construction details for those particular elements must be linked to the model. This happens in Group 5 of the script and the logic behind is similar to a nested dictionary. Each rule check may add a particular set of new elements, and each element has a list of codes associated to it. These codes represent both the BOQ calculation and set of construction details. In other words, this reflects the Retrofit Breakdown Structure explained at the beginning of the chapter, where a retrofit solution has cost and construction information linked broken down for a particular element. The strategy we used to link this information was through custom project parameter stored under the Structural Column category.

A project parameter was created for each new element category. This parameter will record any new component placed in the retrofit phase by filing in with the construction detail code of each element. So, for instance if the Door Confinement Script created a new tie column the corresponding construction will be copied into the parameter called Construction Details.
Group 5 of the Dynamo script: It copies the construction detail codes associated to each element created in the retrofit phase.

This process occurring in Group 5 defines which particular details go with each new element. Now, this relationship ensures that even if the retrofit design is done by independent segments, at the end of the process the sum of all the constructive details and elements will equal the case-by-case design that is fully compliant with national building codes. Once all the rule check scripts for the house category are run, the result can be seen in the Project Information window in Revit:

Resulting codes of the series of Prescriptive Design Rule Checks, performed for a one story masonry house.
The result of all the 6 Main Groups of the Dynamo script include the newly created tie columns, the set of construction details for ring beam rebar placement, anchoring the tie column to the slab and structural detailing of the tie column. Then, to include the cost estimate the Group 6 extracts all elements stored in the Selection Set and gets volumes, lengths and areas depending on each element, to fill in the values of the Bill of Quantities. In the BOQ form, the prices of materials, labor and equipment are preloaded for each city of Colombia. The following images show the resulting details and a screenshot of the BOQ in Revit.

In the left: columns structural detailing including rebar shapes and anchoring to the slab. In the right: Bill of Quantities for a one story masonry house, including description of each item, unit cost and total cost of the intervention, colored green the Door Confinement script corresponding items.

To complete designs of a one story house made up of unreinforced masonry (Category 1 in the Prescriptive Design Methodology), 13 individual rule checks must be run in Dynamo. The resulting structural design is code compliant and can be performed by anyone with training on running Dynamo scripts. At the end of this workflow, a structural engineer has to check the prescriptive designs and approve them to begin the building permit processing.

With the automation of prescriptive retrofit methodology, we managed to reduce the time spent in analysis, design of the intervention and document production from a week and a half to 1 or 2 days depending on the complexity of the house. Having addressed the bottlenecks of the traditional workflow, the new Automated-BIM methods improved the way in which we produce designs. With a fully integrated workflow using 3rd party apps, web-platforms and automation tools we developed a thorough yet agile solution to the National Government for a nationwide implementation.
Next Steps: Taking technology into the construction site

Along with the technical complexity of data collection and design, house retrofitting involves other challenges. First, the number of households that are currently at risk of collapse is massive. Added to the size of the problem, location is another determinant factor. In most developing countries, such as Colombia, Nepal, Philippines and Iran, vulnerable houses can be found as far as the geography goes. Access to sites located along deserts, mountain ranges and valleys can scale up the cost of each step in the value chain of retrofitting. Another critical challenge is the lack of qualified labor for earthquake-resistant construction in the regions where informal housing is common. At the end, all retrofit efforts should be materialized in new walls, columns, beams and real structural elements made from real-life materials, and without qualified construction labor, the designs might just stay in the paper. The combination of this challenges made evident how fragile retrofitting projects – and even national house improvement programs- can be if they are not carefully planned and structured.

To bulletproof the retrofitting workflow, we decided to step out of the comfort of BIM models. In order to ensure that designs are converted into safe structures for people, quality assurance should be present in the site along with quality construction practices. Build Change’s solution to such a complex problem is based on technology and people. We want to reach everyone that has a phone and needs assistance in the construction/retrofit of a safe house. Our idea was inspired in taking the knowledge of a construction supervisor (e.g. architect, engineer or trained technician) and put it somehow in a mobile phone. This way, the number of construction supervisors would be scaled enormously around multiple locations. The tool that we are currently working on is called ISAC (Intelligent Supervision Assistant for Construction). ISAC will be available for homeowners, builders and government employees to check on the quality of the most common structural elements found in informal housing contexts.

Training a mobile phone on good construction practices

The most common construction items found in informal housing contexts are masonry walls and reinforced concrete elements. In most cases, the lack of quality in the construction of these two items leads to a vulnerable structure to earthquakes. Hence, walls and elements with rebar inside were the main target of ISAC and the first step to take technology into a retrofit project.

In order to find any potential issues in construction from a phone, we had to first train it to understand visual deficiencies. This mean to find patterns that make any particular construction scenario compliant with earthquake-resistant practices or not. We call this the Go or No Go criteria. For example, when a wall has proper sized bricks, a running bond with 2.5 mm of mortar between them and no visual damage, we say the wall is a Go. But if a masonry wall has coinciding mortar joints, has a damaged masonry unit or lack of mortar joints we classify it as a No Go. The same principles apply to rebar shapes such as the ones in columns and beams, also referred to as stirrups or ties. There are certain lengths, angles and hooks that need to be present in a stirrup to make it compliant with the building code. Both masonry walls and rebar shapes share the same features that make them compliant or not and these can easily be spotted by a visual check. This means they were the perfect starting point for ISAC to be developed.

After identifying the visual checks, the next step was defining the method used by the mobile phone to perform the quality assessment. The right one for these kind of work turned out to be a
Machine Learning system trained on a large set of examples, indicating the features that make a Go or No Go. Using AI sounds like the right path to take nowadays, when it is present pretty much everywhere you look. However, this imposed a challenge in obtaining thousands of real photos of “good” and “bad” walls or rebars. The process of gathering such a robust training set would be time consuming and involve a tremendous manual input to classify them into Go or No Go status. Here’s where BIM and 3D parametric modeling came in to help and make this ambitious dream possible.

Bridging the gap using Dynamo. A real-life common brick wall being built on the left and an artificial set of BIM-generated images on the right.

Revit and Dynamo served as the ideal environment to create a set of synthetic images that would portray the same characteristics that real life walls and rebar had. By having control on dimensions, colors, angles, shapes and lines we were able to recreate thousands of BIM-generated images resembling walls and stirrups that were automatically classified as Go or No Go. In the image below, we have the script that made possible the creation of more than 10,000 iterations of both good and poor-quality stirrups. The parameters that defined the creation of one sample were length of each segment, angle between each segment, hook length and diameter of each bend in the rebar.
Dynamo script used to generate over 10,000 generations of Go and No Go rebar stirrups, commonly used in columns, beams and tie elements.

Following the training set generation, the Machine Learning was ready to be trained using all of this visual data generated through BIM. In order to achieve maximum accuracy in the results, some visual analysis had to take place. Methods ranging from “U” net analysis to Gaussian blurs helped ISAC to determine if a real photo was a Go or No Go. The framework used to train the Machine Learning of ISAC was the same for both rebar elements and walls. A summary is showed below:

General framework used to train the Machine Learning built into ISAC for Construction Supervision.
The development of ISAC is a pioneering application of AI + Machine Learning powered by BIM to make construction supervision accessible to millions of people living in vulnerable houses. The app is currently in an early deployment with the government of Colombia, framed inside the national house improvement program “Casa Digna, Vida Digna” and will be tested internationally as well. Our goal with this kind of research projects is to explore the potential benefits and disruptive gains that we can get from BIM, AI and ML oriented towards saving lives. By following this path and expanding the use of technology on solving substandard housing and the risk associated to the built environment, Build Change aims to be a pioneer in new ways of building, assessing, designing and supervising safe houses worldwide.

First results of the ISAC app. The images on the left show a poorly built rebar shape that scored a No Go status, while on the right the properly-built stirrup is classified as Go.

Conclusion

This case study showed how BIM, automation and 3rd party app integration transformed an existing workflow for retrofitting informal housing into a simple and efficient one, that can be scaled at a national level. By removing bottlenecks from the traditional methods, we achieved not only cost and time improvements, but we also converted an obscure methodology with limited practitioners into a simple yet thorough framework of automation tools. The new workflow relies on automation and prescriptive engineering to rapidly produce code-compliant designs in less than 2 days.

Build Change has pioneered the use of Building Information Modeling into Disaster Risk Reduction initiatives. This implementation has opened a range of possible uses of technology from the AEC industry such as laser scanning, virtual reality, Blockchain and Artificial Intelligence.
In essence, this case study aims to demonstrate the benefits of tech-based workflows to tackle an issue that affects millions around the globe. Disaster Risk Reduction should be as efficient as private and corporate projects, given the fact that its main focus is reducing deaths from natural hazards. This case study focuses on a particular use of BIM technology to solve substandard housing in Colombia, but gathered experience from post disaster and prevention programs in Nepal, Philippines, Haiti and Sint Maarten. We will continue to innovate and promote the use of technology to mitigate risk in an efficient, scalable way.

Thank you for your attention and interest in this session.

Nicolas Ortiz Abello
nicolas@buildchange.org

Allie Young
allie@buildchange.org

Andres Felipe Robles
andres@buildchange.org

Sebastian Moreno
sebastian@buildchange.org