AUTODESK. UNIVERSITY

AS320122

BIM View: Proactive Model Management Stop Reacting - Start Responding

Holger de Groot
National Director of BIM (HDR)

Mehdi Blanchard
Technology Innovation Specialist (HDR)

Learning Objectives

- Explain why visualizing data can help you to proactively manage your Revit models.
- Understand the benefits of a model integrated “health check family” for quality control.
- Understand the benefits of a web-based dashboard solution to visualize performance.
- Understand the pros and cons - what we learned during the development process.

Description

BIM is a process and understanding the process and managing it accordingly is the key to a successful BIM implementation. But how do you measure the development of a building information model? Do you have methods in place to understand when the potential for issues on your models occur? The best way to accomplish this is through diagnostic tools that alert users, when certain pre-defined rules are broken, or limits are exceeded.

In this session, we are presenting the lessons we learnt through the development of the ‘BIM View’ tool – a macro-based model health check solution, combined with a web-based dashboard to visualize the quality and performance of all our Revit models. Learn what worked and what did not and why we replaced Dynamo with a macro based solution, and Power BI with a combination of Vue.js and Data Driven Documents (D3.js), gathering data directly from all our active projects and pushing it into a web based dashboard to proactively manage all our models.

About the Speaker

Holger de Groot re-joined HDR in 2017 and brings a wealth of knowledge and experience in Building Information Modelling & Management, Project Management and Healthcare Design to the Design Technology Leadership Team. As National Director of BIM for the Australian region, his role is to supervise and guide Digital Practice Leaders at HDR. He is responsible for implementing and advising leadership on the corporate Digital Practice BIM Strategy, interacting with various disciplines and advising on BIM matters at all levels.

Holger is not only passionate about the implementation of BIM in the planning process, but he also has a strong link to the construction side of projects, having participated in and managed BIM projects in a variety of roles in Australia, New Zealand, Germany, and the United Kingdom. Holger is also a sessional lecturer at the University of NSW, registered at the Institute of Architects (#20262) in Germany (AKNDS), and certified BIM Manager with BIMcreds (Design - Managerial Role). Holger is also a sessional lecturer at the University of NSW, registered at the Institute of Architects (#20262) in Germany (AKNDS), and certified BIM Manager with BIMcreds (Design - Managerial Role).

About the Co-Speaker

As Technology Innovation Specialist and Software Developer at HDR for more than 7 years, Mehdi Blanchard's role is to assess new technologies and find creative ways to implement them in HDR's workflows. He also researches, designs and develops custom software solutions for HDR globally to improve the BIM process. Mehdi has worked in the AEC industry for more than 15 years. He holds a postgraduate degree in software engineering from the Mediterranean Institute of Research and Computer Science and Robotics (IMERIR) in Perpignan, France and has also studied computer graphics and 3D animation at the Computer Graphics College of Sydney (now SAE).

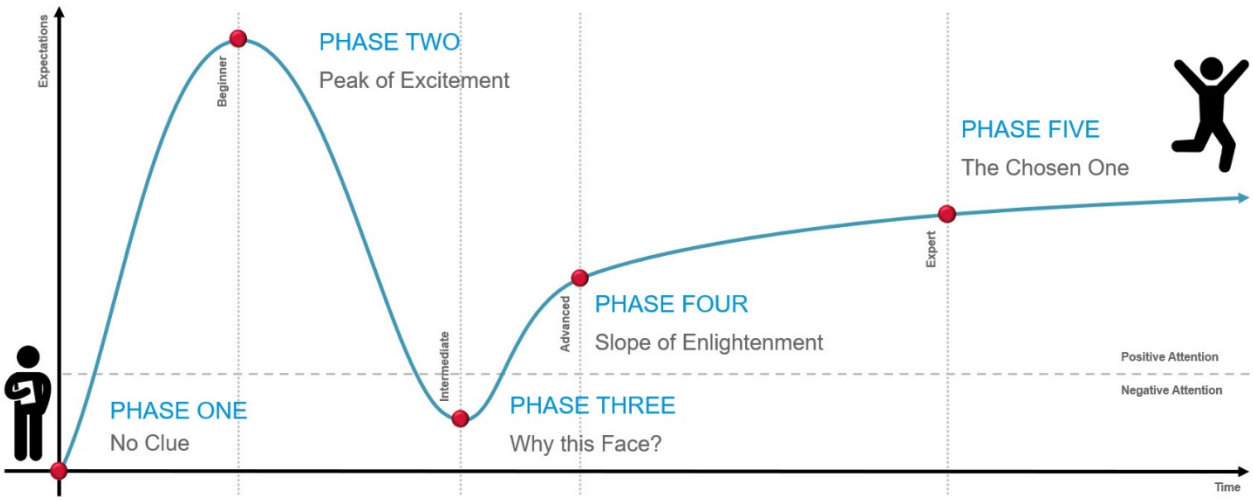
Contents

Stop Reacting

One of the things that occur in many offices is that, after training, everyone is expected to be an expert at using Revit. But it takes time for new Revit users to get comfortable with the program, the new workflows and to build up a high level of user knowledge.

New Revit users are usually being asked to learn the software as they are working on a project. Even if they come prepared with basic training, they will quickly find that not everything is going to work like it did in their training sessions which can lead to frustration!

Quite often, new Revit users not only have to educate themselves online (webinars, eLearning, user forums, etc.) but also find themselves without the support they need to be able to review what they couldn't do and everything else that caused frustration. Revit's inability to provide user-friendly feedback isn't making it easier for them.

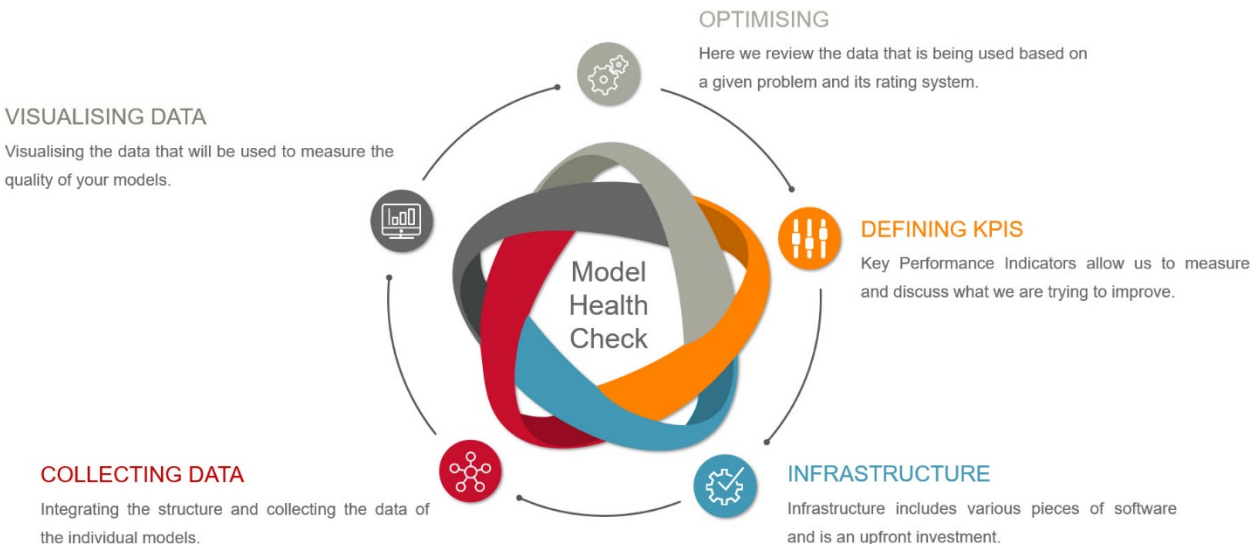


Five Phases of Revit

We realized that it was time to develop a tool for our Project Leaders and BIM Managers that could not just be used to monitor the project status and 'health' of individual models, but would also allow them to identify critical areas before they turn into problems – a tool that would allow them to proactively manage a project.

Start Responding

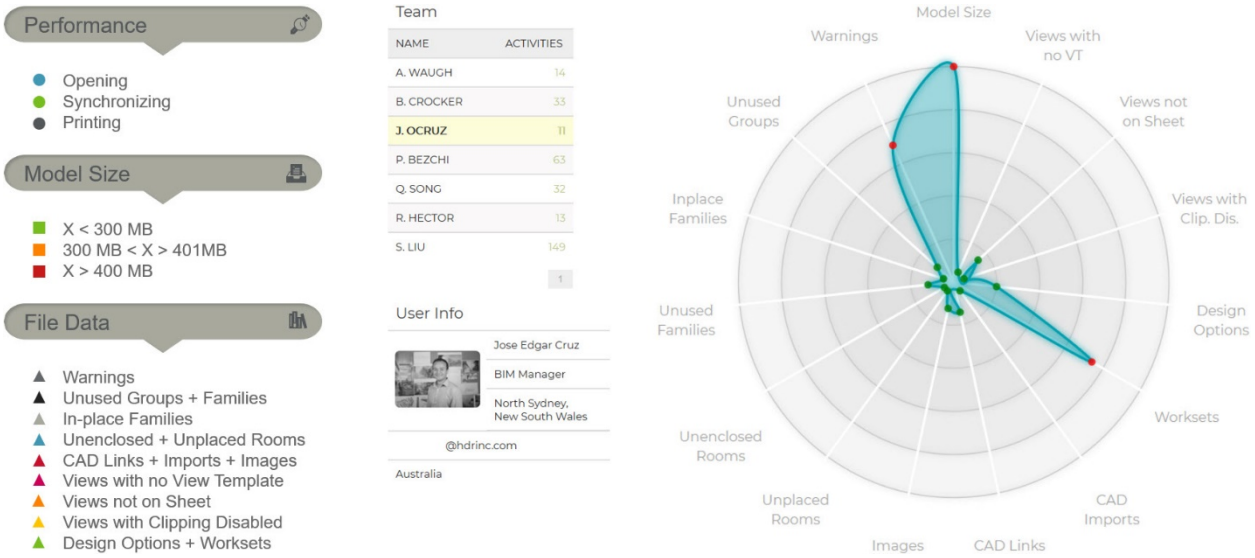
Building information modelling is a process and understanding the process and managing it accordingly is the key to a successful BIM implementation. But how do you measure the development of a building information model? Do you have methods in place to understand when the potential for issues on your models occur? The best way to accomplish this is through diagnostic tools that alert users, when certain pre-defined rules are broken, or limits are exceeded. When we started to develop the 'BIM View' tool, we followed a five-step strategy.



Model Health Check Process

Defining KPIs

A Key Performance Indicator (KPI) is a quantifiable metric that reflects how well a user, project team or office is achieving its stated goals. For example, if one of your goals is to keep the amount of warnings within your models below a total of 400, you could use a KPI to target the number of warnings that remain unsolved. This will measure the team's progress toward your goals.



Key Performance Indicators

Used well, KPIs support your goals and strategy. They allow you to focus on what matters most, and to monitor your progress. At the beginning, we choose KPIs that were directly linked to the model's health to measure how well our models performed against our key goals.

S.M.A.R.T.E.R

A way to evaluate the relevance of your KPI is to use the SMARTER framework:

- **Specific:** be clear about what each KPI will measure, and why it is important;
- **Measurable:** your KPI must be measurable to a defined standard;
- **Achievable:** your models must be able to report on the KPI;
- **Relevant:** your KPI must measure something that matters and improves performance;
- **Time-Bound:** the target must be achievable within an agreed time frame;
- **Evaluate:** continuously evaluate your KPI to improve the overall process;
- **Re-evaluate**: continuously reevaluate your KPI to optimize the overall process.

Infrastructure

Once we were satisfied that we had meaningful KPIs to measure the health of our models, we started to look at the infrastructure. The infrastructure can include various pieces of software and is an upfront investment that will allow you to collect data from individual models. We reviewed multiple solutions before we were able to find the one that was suitable for our workflow.

Revit > Microsoft Excel

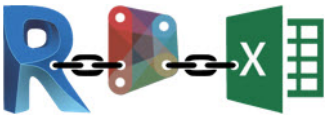
When we started, the auditing process for models included excel spreadsheets that were manually filled with data – a workflow that did not meet our requirements:

- **Con:** Data input 100% manually
- **Con:** Needs to be scheduled
- **Con:** Time consuming

Revit > Dynamo > Microsoft Excel

We improved this workflow by developing and using a Dynamo graph that partly automated the data input – an improvement but it did not meet our requirements:

- **Pro:** Data input partly automated
- **Con:** Does not start automatically
- **Con:** Dynamo version control needed
- **Con:** Warnings via HTML (Revit 2017)



Revit > Dynamo > Dynamo Player

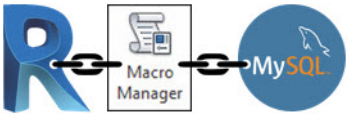
In the next step, we moved away from excel spreadsheets and developed a ‘model health check family’ that was automatically filled with the data, using the dynamo player – a massive improvement but it did not meet our requirements:

- **Pro:** Warnings directly available
- **Pro:** Data input 100% automated
- **Con:** Does not start automatically
- **Con:** Dynamo version control needed
- **Con:** Graph takes 1½ Minutes / Model

Revit > Macro Manager > MySQL

Finally, we found a solution that met our requirements by using a macro in combination with MySQL. This solution allowed us to fully automate our infrastructure:

- **Pro:** Starts automatically
- **Pro:** Data input 100% automated



Collecting Data

After the infrastructure was implemented, we were able to start the data collection for each KPI within our models. But first, we had to define the data collection frequency. Therefore, we looked at the data that wanted to be collected continuously while others could be collected on an hourly, daily, or monthly basis. We decided that we want the macro to run every four hours, minimizing the impact on the users experience while working with Revit. Also we run this process when the synchronization is completed to make sure we have the latest data. The code below breaks down the different steps involved in the macros.

Macro Process Overview

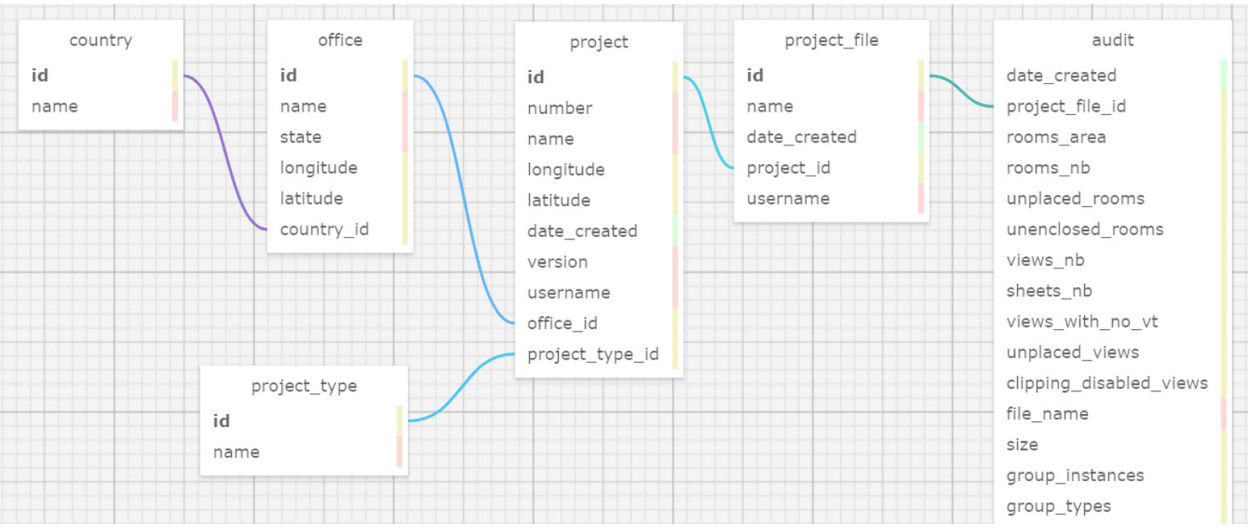
1. Register event when a Revit user opens, synchronises or prints the local model;
2. Extract the data required for the audit and create a timestamp to measure time taken during opening, synchronising or printing;
3. Send the data to the auditing web service. The PHP based webserver records the data in the MySQL database;
4. Find the 'Model Health Check' family in the model;
5. Update the 'Model Health Check' family with new acquired data.

```
// 1. Register event when user synchronizes the document
this.Application.DocumentSynchronizedWithCentral += new EventHandler(Application_DocumentSynchronized);
// 2. Get the data when the user synchronizes
FilteredElementCollector collector = new FilteredElementCollector(doc).OfClass( typeof( SpatialElement ) );
foreach(SpatialElement e in collector) {
        // count number of rooms, room area, check if room is placed, …
}
// 3. Post data to the webserver
var request = (HttpWebRequest)WebRequest.Create("https://myauditserver/audit.php");
var stream = request.GetRequestStream()
stream.Write("mode=AuditFile? rooms_area =roomArea &rooms_nb = roomNb", …);
var response = (HttpWebResponse)request.GetResponse();
// 4. Find the Health Check family in the project file
IEnumerable<FamilyInstance> instances = new FilteredElementCollector( doc ).OfClass( typeof( FamilyInstance ) ).Where( x
=> x.Symbol.Family.Name.Equals("Health Check family" ) );
// 5. Update the Health Check family
Parameter param = instance.LookupParameter("Number of Rooms");
param.Set(roomNb);
```

Macro Process Overview

Database Schema

The database schema is actually pretty simple. We store the latitude and longitude of offices and projects to be able to display them on a map. All the audit data can be stored in a large table or grouped in smaller tables.



Database Schema

We currently use a MySQL database and we use views to limit the data we pass to the web viewer to have a more responsive interface. For example, we display only the projects that have been active in the last two weeks, making sure that the data we see is current. So we create a view called 'recent_projects' that limits the projects we want to see as follows:

```
SELECT project.id as
FROM project
INNER JOIN project_file ON project.id = project_file.project_id
INNER JOIN audit_activity ON project_file.id = audit_activity.project_file_id
WHERE (audit_activity.date_created > DATE_SUB(now(), INTERVAL 14 DAY))
```

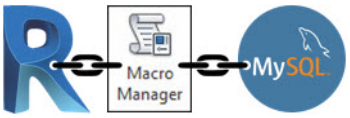
Visualising Data

You can significantly decrease the time it takes to compare metrics by compiling them into one place with an intuitive, visually-appealing format. Dashboards can combine the data that has been extracted from multiple models and communicate this information to the entire team, visualising the performance of their individual models. When designed correctly, dashboards provide an extraordinary level of accessibility and efficiency. This greatly improves the decision-making process, by providing you and your team with the right data to make an informed decision.

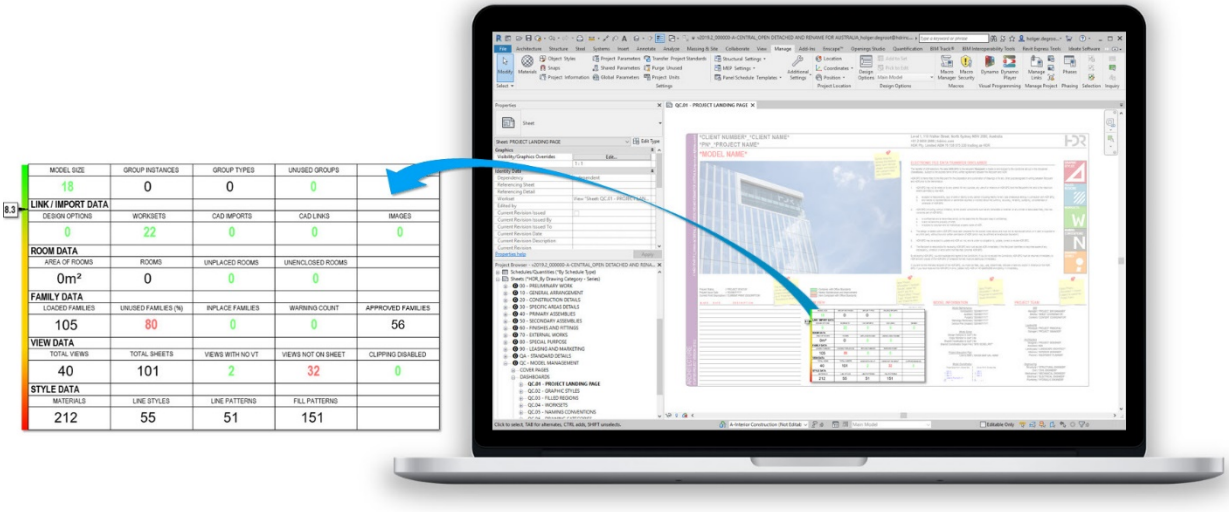
Revit – Project Landing Page

There are different approaches and opinions to what a landing page should include. However, having a 'Model Health Check' family imbedded has significantly increased the visibility of our KPI's and their impact on model performance:

- **Pro:** Starting View (High Visibility)
- **Pro:** Data input 100% automated
- **Pro:** Updated every 4 Hours



- **Con:** Only accessible in Revit
- **Con:** Snapshot of the model state (no history)
- **Con:** Doesn't allow comparability



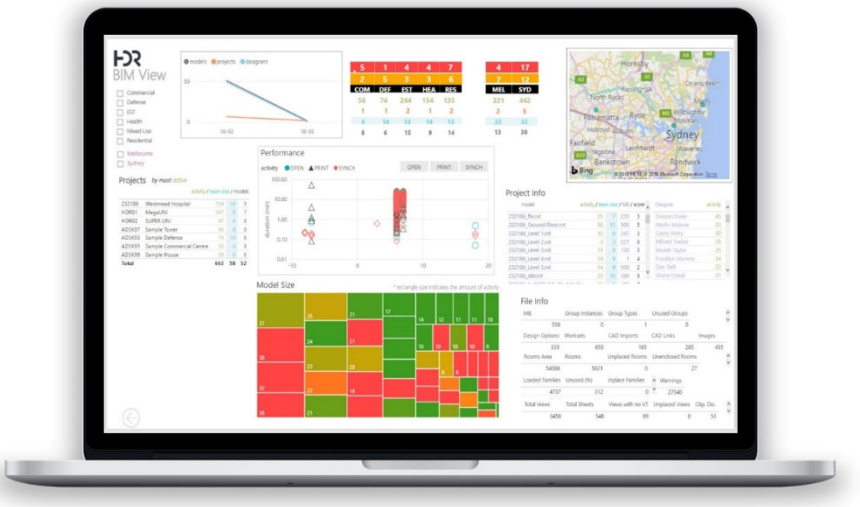
BIM View - Model Health Check Family

Power BI – Dashboard

Microsoft Power BI is basically Excel pivot tables with Excel's data visualisation tools taken to the next level. Its user interface is very bulky and depending on the device and browser that we used, it had a negative impact on the visibility and usability of our dashboard:

- **Pro:** Transforms data into rich visuals
- **Pro:** Data input 100% automated
- **Pro:** Accessible to everyone
- **Pro:** Allows comparability

- **Con:** License costs (maintenance)
- **Con:** Compatibility issues (devices)
- **Con:** Needs training and support



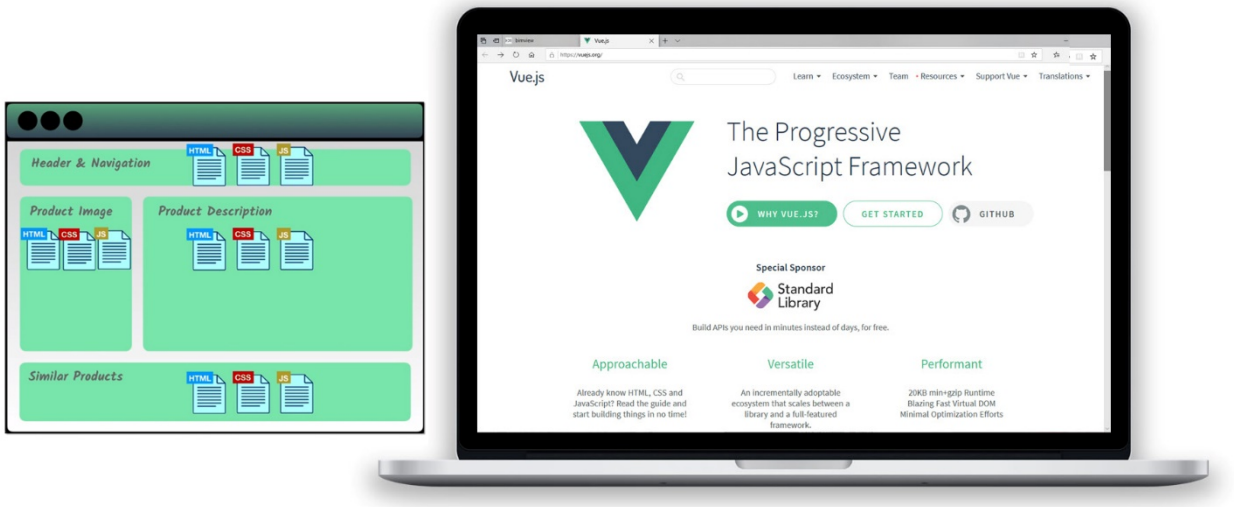
BIM View – Power BI Dashboard

Vue.js – Model-View-Viewmodel

Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable – which was perfect for us in combination with D3.js:

- **Pro:** Easy to pick up
- **Pro:** Reactive
- **Pro:** Integrates with other libraries

The web page is split into as many components as needed:



Vue.js – Example of Components Layout (R) and Website (L)

Each component has its own HTML, Javascript, and CSS. Which makes maintenance a lot easier as the files are a lot smaller and can be reused as needed. The local CSS in one component doesn't interfere with the CSS in another component. See below an example of a Vue.js component structure:

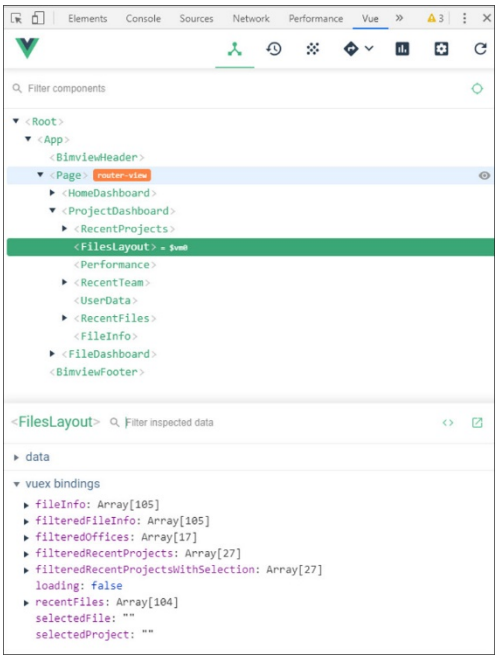
HTML	 {{ project }}

Javascript	export default {
	name: 'bimview-header',
	data: function() {
	return {
	project: 'Westmead' }
	}
	}

CSS	<style lang="scss" scoped> #name { color: #fff; } </style>

Vue.js – Component Structure

Vue.js is **reactive** which means that if the value of the variable "project" changes, then the displayed value will automatically change as well. For example, if you use a filter based on a country, then all the graphs and texts will automatically be redisplayed when the country changes. It makes the development very efficient and the web interface dynamic.



Vue.js – Inspector

Data can also be stored in a global store that all the components can access. An add-in can be installed in Google Chrome to access / modify all the data while running in development mode, which makes development a lot easier, as you can also track events.

D3.js – Data-Driven Documents

D3.js is a JavaScript library for manipulating documents based on data. It helps users to produce dynamic, interactive data visualizations in web browsers. Combined with Vue.js, it was the solution we were looking for to produce our BIM View dashboard:

- **Pro:** Transforms data into rich visuals
- **Pro:** Flexible and easy to understand
- **Pro:** No license costs
- **Pro:** Data input 100% automated
- **Pro:** Accessible to everyone
- **Pro:** Allows comparability
- **Pro:** Vector graphics
- **Pro:** Graphics can be animated

D3.js – Data-Driven Documents

Examples of graphics that can be achieved with D3: https://github.com/d3/d3/wiki/Gallery

We looked at the graphs in the gallery and tried to figure out which ones could help us achieve the best visual experience:

1. Create an SVG element in HTML;
2. Load Data or subscribe to an event when the data changes;
3. Set up XY Axes;
4. Iterate over data to build SVG Graphical Elements.

Steps	Code
Create an SVG element in HTML	var svg = d3.select("body").append("svg") .attr("width", width + margin.left + margin.right) .attr("height", height + margin.top + margin.bottom) .append("g") .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
Load Data	d3.tsv("data.csv", type, function(error, myData) {
Set up XY Axes	x.domain(myData.map(function(d) { return ….; })); y.domain([0, d3.max(0, (myData,…)]); svg.append("g") .attr("class", "x axis") .attr("transform", "translate(0," + height + ")") .call(d3.svg.axis().scale(x).orient("bottom")); svg.append("g") .attr("class", "y axis") .call(d3.svg.axis().scale(y).orient("left"));
Iterate over data to build SVG Graphical Elements	svg.selectAll(".bar") .data(myData) .enter().append("rect") .attr("class", "bar") .attr("x", function(d) { return x(d.letter); }) .attr("width", x.rangeBand()) .attr("y", function(d) { return y(d.value); }) .attr("height", function(d) { return height - y(d.value); });

Vue.js – Inspector

The D3 library is now split into individual nodes, which means you don't need to load the entire library, just use the nodes you need inside each Vue component.

Optimising

In the final step, we reviewed the data that was collected on a given problem and its rating system. Where needed, we re-evaluated KPIs to optimize the overall model health check process. Once this evaluation was completed, we deployed the 'BIM View' tool for the Australian region. Now, we are in the process of deploying it on a global scale across HDR in multiple languages.

BIM View – Dashboard: Home

Proactive Model Management

- Projects with dedicated BIM Managers vs without (Performance vs. Performance)
- Usage of approved and non-approved families (Approved Family Parameter)
- Improving Model Performance (Model Size and File Data)

BIM View – Dashboard: Projects

Proactive Support and Training

- Comparing teams to find our champions (Mixing High and Low Achievers)
- Providing help before a support request is being launched (Monitoring and Controlling)
- Allowing users to see the impact of their own actions in real time (Model Rating 0 - 10)

BIM View – Dashboard: Files

Lessons Learned

Having an automated model health check tool such as the 'BIM View' in place helps us and our BIM Managers to proactively manage our models. However, as a team model health shall always be everyone's responsibility – a BIM Manager alone cannot fix it all.

We also realized how important it is to provide a dashboard that is highly visible – not just to the individual Revit User and their BIM Manager but also to our Project Leaders who can now 'see' which positive effect a dedicated BIM Manager can have to their project.

Therefore, we encourage our BIM Managers always to keep an eye on the model health check metrics, so they know when help is needed before a support request is being handed in by a Revit user. That allows them to proactively manage their models while improving the performance.

Acknowledgement

Since Mehdi and I started to work on the 'BIM View' tool, we have received support, comments and advice from many professionals in the AEC industry from around the world. Without them, our work would have been harder and therefore, we would like to take this chance to thank them for their input:

- **Mark Schoolman** who introduced us to the existing auditing system and its shortcomings;
- **Josh Moore** and **Justin Benjamin** who shared their custom annotation family with us which we used as a prototype to develop our first 'Model Health Check' family;
- **Ron Croke** whose 'Health Check' Dynamo graph helped us to develop our own graph;
- **James Wright** who developed the first 'Model Health Check' family and Dynamo graph;
- **Konrad Sobon** whose 'Web-based Project Management' presentation gave us new ideas;
- **Dan Chasteen** and **Nicholas Cameron** whose 'You Can't Manage What You Don't Measure' handout helped us developing our first version of the 'BIM View' dashboard;
- And last, but not least **David Austin** and **Simon Leith** for their great IT support.

Knowledge sharing is an important part of what we do in our daily work and events like the Autodesk University also foster that sharing.