BO3158 – The Business Proposition for Developing Custom Tools

Robert Manna & David Spehar
BIM R&D Leader / BIM Leader - Stantec
In the era of 2D CAD, many firms invested in custom tools and development staff to write and maintain them. A firm's overall experience varied in terms of the success of this investment and just about everyone has a story to tell about being "burned" from spending money developing custom tools. However a number of market forces are coming into play that continue to drive the need to customize the Autodesk® Revit® software platform, and the question is, "Do you rely entirely on third-party software or go back down the road of customization?"

This class explores the benefits of investing in the development of custom tools for your firm, and shows how Revit is really "different this time" for those who have a horror story to tell. This class looks at the tools Stantec has developed in the last year and the return on investment we have already seen. We also examine our overall strategy and approach to development.
Key learning objectives

At the end of this class, you will be able to:

- Convey the value of custom tools for Revit
- Explain the difference between tools and utilities in Revit versus CAD
- Develop an internal strategy for development
- Review resumes of potential candidates who develop Revit tools

- We are not going to talk at all about how to program or what you can or cannot do.
I want....
API What?

- Application Programming Interface (API)
- Uses the Microsoft dot framework (.Net)
- Can write code in Visual Basic or C# (see-sharp)
- Access to other applications and the Operating System
Why develop custom tools

- Autodesk is not going to do it:
  - Revit has become a platform, similar to AutoCAD or Office.
  - There is a lack of competition in the market.
  - Autodesk would/will never be able to address every need.
- You know your own processes.
How is this different from CAD…?

- Revit provides a set infrastructure in the form of categories
  - Cannot create, alter or delete categories
- CAD was dependent on layers
  - Users easily deviate from layer standards
- Software is updated annually now, easier to “keep up”
Know what is on the market

- Re-invent the wheel only when it makes financial sense.
- Know what features you need, and which ones you do not.
- Contact App developers (no one is getting rich on AEC apps)
Estimating Development Cost

- “Simple” is easy
- More features increases risk;
  - Complexity of core code
  - More features equals more UI
- Fixed fees, even when conservative will always be wrong.
- Plan on a contingency
  - Bugs will always show up
Its easy to dream up “simple” yet complex tools

- I just want a tool that creates a “sketch”:
  - Can it also number the sheet for me?
  - Can we classify the sheet?
  - What about automatically cropping the view?
  - What if…. 
Dreaming
Minimum Viable Product (MVP)

- Goal is to keep it simple.
- What is the minimum you need?
- Plan to make it more complex through iterative development cycles.
Keep It Simple Stupid (KISS)

- Most of our tools are simple, limited UI
  - Click button, pick source file
Keep It Simple Stupid (KISS)

- Most of our tools are simple, limited UI
  - Click button, pick source file
- Some are more complex
  - Multiple options
  - Multiple dialogs
Keep It Simple Stupid (KISS)

- Most of our tools are simple, limited UI
  - Click button, pick source file
- Some are more complex
  - Multiple options
  - Multiple dialogs
- Look for common behavior
  - Importing data from Excel
Planning
What to develop?

- Create a plan (roadmap)
  - Often necessary to justify the investment
What to develop?

- Who are your users?
- Where are their pain points?
- What is their skill level?
- Different constituencies
  - Engineers/Architects
  - Designers/Production Staff
- You cannot make everyone happy, if no one is satisfied, you did your job.
What to develop?

- We had a list of candidate tools
- Worked with our developer to review
- Attempted to stick to “Easy” & “Medium”
Planning

- Other logistics to think about.
- How will you:
  - Develop your User Experience (UX); organize Tabs, Panels, Split Buttons, etc.
  - Deploy the software and manage updates & bug fixes?
  - Manage versioning for Revit?
  - Beta Test software?
Planning

- Help Documentation
  - Clear and concise
  - Easy to update (not a help file)
  - Organized
  - Feedback link/button (built-in)
Executing
Organize

- Prioritize
  - Value to user/practice (return)
  - Cost to Develop (effort)
- Combine the two and reach a “Development Priority”
- This was not a scientific process.
Paying for development

- To fund development you need a plan:
- Tool X will cost: $1000
- Tool X can save our users: 10 minutes each time
- Users will use Tool X: Once per day

One User:
10 minutes $\times$ 250 days $= 2500$ minutes
2500 mins/60 mins $= 41.6$ hours
41.6 hrs $\times$ $75$ rate $= 3120$
$3120 - 1000$ dev cost $= 2120$ annual savings
Our ROI Graph

<table>
<thead>
<tr>
<th>Tool Name</th>
<th>Man Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAD Manager</td>
<td></td>
</tr>
<tr>
<td>Room Placer</td>
<td></td>
</tr>
<tr>
<td>Key Schedule Manager</td>
<td></td>
</tr>
<tr>
<td>Sheet Page Number Update</td>
<td></td>
</tr>
<tr>
<td>Room Generator</td>
<td></td>
</tr>
<tr>
<td>Sketch Creator</td>
<td></td>
</tr>
<tr>
<td>Shared Parameter Creator</td>
<td></td>
</tr>
<tr>
<td>Sheet Generator</td>
<td></td>
</tr>
</tbody>
</table>

Projected labor savings based on an average project using all tools

Total Savings (hrs) = 150 x ave rate $ 95.00 = $14,220
SUMMARY ROI PER PROJECT

- Stantec Tools Workflow
- Revit Default Workflow
Paying for it

- Encourage Projects to pay for development
- Potential to be “deployed”
  - Depends on level of finish
  - Value to other teams
- It is all still the company’s money
  - Avoid territorial claims up front.
  - The tool belongs to the company

\[
\begin{align*}
20 \text{ models} \\
\times 6 \text{ minutes} \\
\times 2 \text{ hours} \\
\times 50 \text{ weeks} \\
\times 100 \text{ hrs} \\
\times 75 \text{ rate} \\
= 7500
\end{align*}
\]
Executing the work

- You need a developer
- This is more than AutoLISP
- C# & VB are both full languages for writing applications
- Familiarity with data structure can be helpful.
- Experience counts!
- Creativity & thinking skills
Finding the right person

- Things to consider:
  - Internal or External?
  - Which attributes are most important?
    - Database experience
    - Application Development
    - Industry Experience
  - “Ownership” of final product
  - Intellectual Property Rights
    - very murky and grey area when it comes to software and code
Executing the work

- Clear chain of command
  - Our developer reports to “me” (BIM R&D Leader)
- No different than any other project.
  - Watch the budget.
  - Assure goals can be achieved
  - Make the tough decisions when you have to, “we have to stop here/now”.
  - Feedback Loop(s)
Justification & Execution

- How do you assure you receive fair value?
  - Start simple, short term project/contract.
  - From the roadmap identify one or two easy projects.

- Do the math for leadership:
  - If we develop Tool X our return on investment will be $x

- User feedback helps.

“I just want you to know that this is FANTASTIC!! this saved a few days of my time.”

“because of your help…I now have all of the required parameters loaded into the models”
Conclusion

- Customization makes sense for many different reasons
  - To do something the software cannot currently do.
  - To alleviate users of repetitive low value tasks.
  - To eliminate error and improve quality.
  - A combination of the above.
- Find the reason that makes sense for you, build a case
Please fill out surveys.

Questions?