

AS323405-R

Is AEC Computational Design for Everyone?

Timon Hazell
Silman

Learning Objectives

- Gain a broader understanding of best practices for the structure of computational design in differing office environments
- Collectively define what computational design competency should be expected of various roles in our organizations
- Find answers to the questions you have about computational design in your office.
- Understand what professional development paths look like for computational design savvy gurus

Description

One highly debated topic in the AEC industry over the past 5 years is this exact question: "Is Computational Design for Everyone?" Are you struggling with this question? Are you considering which team members need what training? Are you managing the infrastructure of your digital design initiatives? Are you asking if Computational Design should be a separate department or part of every designer's skillset in 2020? Is Computational Design for all designers or a select group? Have you been trying to increase buy-in for computational design across your offices? Join us as we discuss what computational design looks at offices around the world. This will be a heated discussion, so you will not want to miss it and your contribution to the industry will be invaluable to others. As an outcome we will produce a general summary, as a group, that helps define successful computational structures, and maybe discover something completely new in the process.

Speaker(s)

Timon Hazell is a mover, a shaker and all around passionate energetic Building Digital Design enthusiast. He is a highly rated speaker and if you haven't attended one of his sessions before, you are in for a treat! He would tell you to stop reacting to the industry and take your projects by the horns and wrestle them into the future. He will also tell you that no one tool is going to fix your problem. In the past few years he has founded DynamoDC, sat on the board of Revit DC and spoken at conferences across North America. His background working at architectural, engineering, and construction firms gives him a broad foundation that he now puts to use at Silman, a multi-office structural engineering firm across the Northeast. His responsibilities include project engineering, modelling, digital tool development, and training. He is grateful to the open learning environment at BILTNA and grateful to the generous knowledge experts and he seeks opportunities now to give back to the building industry.



Introduction

rules of engagement

(introductions, firm size, role in firm, a person in this industry who has inspired you, 1 question you would like to see answered)

define cd (computational design and levels of users)

Yesterday (let's frame the discussion with facts first)

how have you seen computational design used/implemented unsuccessfully? What made it unsuccessful, what could have made it successful?

what are common misconceptions of computational design?
what pain have you experienced around cd?

provide a specific example of cd being used well/examples of innovative ideas you have seen take off in the last 5 years

who have been the innovators in your office? Other than yourself?

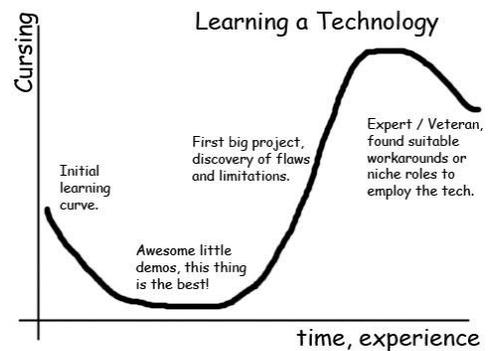


FIGURE 1: STEPHEN CALENDAR BLOG

Today

how many people are using cd at each level in your firm?

compile a list of computational design consultants.

compile a list of firms that we have seen use cd well.

what are characteristics of your cd super users.

where have people learned cd skills.

what are good ways that small disruptions have assisted your teams' projects. What did these require?

what are good ways large disruptions are assisting your teams (give examples) What did these require to gain steam?

what industries can AEC borrow ideas for innovation? Provide Examples.

Tomorrow

what are the risks to implementing/using cd?

what are the benefits of cd? What are the risks?

what questions do you ask before throwing cd at a problem?

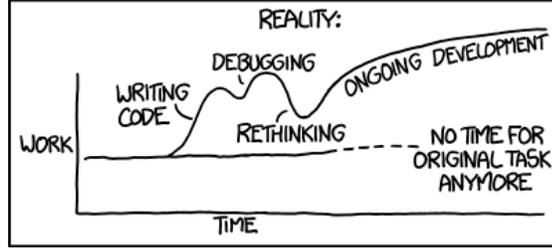
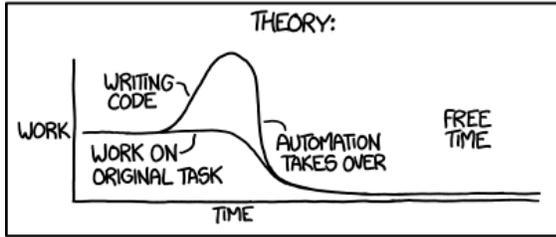
what processes/norms/culture will be disrupted (or needs to shift, adjust, etc.) for a company (or the industry) to embrace computational design? or something like that

assemble a workflow for deciding the importance and urgency of ideas and whether to escalate the idea to a task

assemble a process for communicating to stakeholder potential cd problems and receiving buy-in

Conclusion: Let's summarize it.

"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"



THE SIMPLE ANSWERS
TO THE QUESTIONS THAT GET ASKED
ABOUT EVERY NEW TECHNOLOGY:

WILL <input type="checkbox"/> MAKE US ALL GENIUSES?	NO
WILL <input type="checkbox"/> MAKE US ALL MORONS?	NO
WILL <input type="checkbox"/> DESTROY WHOLE INDUSTRIES?	YES
WILL <input type="checkbox"/> MAKE US MORE EMPATHETIC?	NO
WILL <input type="checkbox"/> MAKE US LESS CARING?	NO
WILL <input type="checkbox"/> DESTROY MUSIC?	NO
WILL <input type="checkbox"/> DESTROY ART?	NO
BUT CAN'T WE GO BACK TO A TIME WHEN -	NO
WILL <input type="checkbox"/> BRING ABOUT WORLD PEACE?	NO
WILL <input type="checkbox"/> CAUSE WIDESPREAD ALIENATION BY CREATING A WORLD OF EMPTY EXPERIENCES?	WE WERE ALREADY ALIENATED

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

HOW OFTEN YOU DO THE TASK

	50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
6 HOURS				2 MONTHS	2 WEEKS	1 DAY
1 DAY					8 WEEKS	5 DAYS

HOW MUCH TIME YOU SHAVE OFF
