

Expanding your
Repertoire

Design Heuristic
One Experiment
at a Time

2 hour hands-on

Rebecca Wirfs-Brock
rebecca@wirfs-brock.com

@rebeccawb
<https://www.wirfs-brock.com/>

© 2023 Wirfs-Brock Associates

1

UMWELT

Umwelt means "environment" or "surroundings: our perceptual world [merkwelt] and everything we do to affect it [wirkwelt].

Our umwelt isn't static; we reshape our umwelt as we interact with the world, gaining skills and learning to perceive differently. It is our umwelt that both drives and limits our potential actions.

2



What is a heuristic?

Billy Vaughn Koen,
*Discussion of The Method:
 Conducting the Engineer's
 Approach to Problem Solving*

“anything that provides a plausible aid or direction in the solution of a problem but is in the final analysis unjustified, incapable of justification, and potentially fallible.”

“any approach to problem solving, learning, or discovery that employs a practical method not guaranteed to be optimal or perfect, but sufficient for the immediate goals.”
 —Wikipedia

3

Three Kinds of Heuristics to Look For



Illustrated by a Conversation with
 Chelsea Troy about Testing

4

1. Action Heuristics

things we do to solve our immediate problem

- “write a test, then write code to pass the test”
- Design patterns
- There are many modeling, architecting, testing and development techniques/actions both smaller and larger than patterns.

5

2. Value Heuristics

determine our attitude and motivate our behavior

- TDD values:
 - tested code.
 - testing is integral to design and implementation.
 - Values determine what actions *seem* appropriate.
 - Write tests before writing code
 - Check that all tests pass before committing code
 - Check in tests with working code

6



3. Guiding Heuristics

suggest related actions

- Chelsea's guiding heuristic for tests:
 - Don't treat test code the same as production code, instead, make each test understandable in isolation.
- leads her to:
 - write self-contained test methods
 - not factor out common code
 - not apply DRY

7

SHOPPING TEAMS

BAD:
TWO NON-NERDS

LET'S GET THAT ONE.

OKAY.

GOOD:
NON-NERD + NERD

LET'S GET THAT ONE.

WAIT, I THINK THE OTHER ONE MIGHT BE A BETTER DEAL.

OKAY THAT ONE.

VERY BAD:
TWO NERDS

HOW ABOUT THAT ONE?

I THINK THE OTHER ONE MIGHT BE A BETTER DEAL...

HEM, I'M NOT SURE...

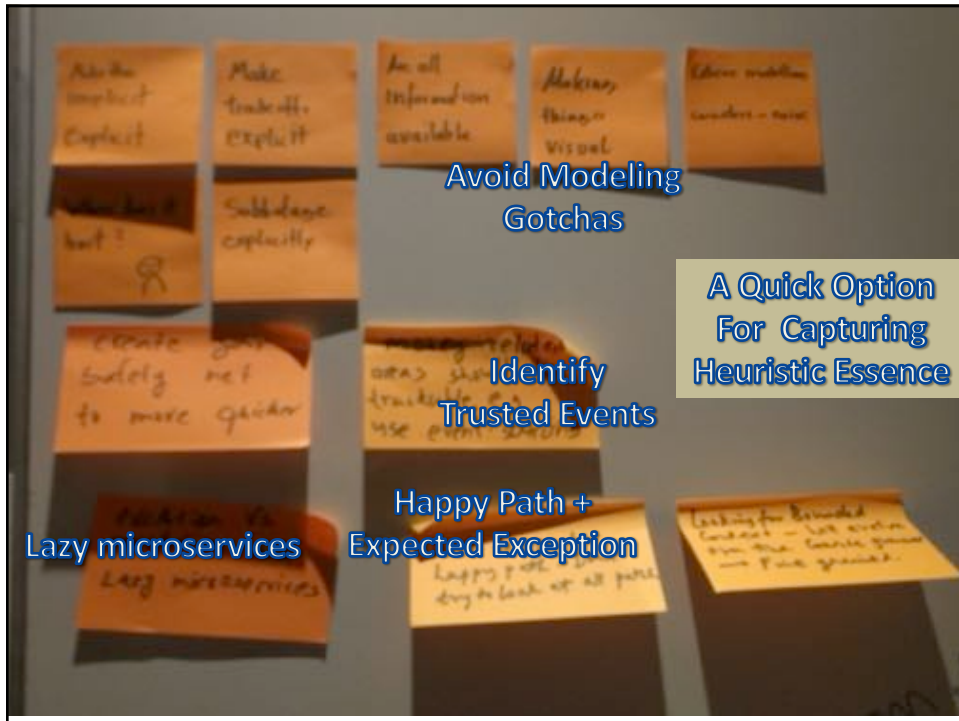
TWO HOURS LATER

I THINK OUR MAIN PROBLEM IS OUR UNCLEAR DEFINITION OF VALUE.

THAT IS NOT YOUR MAIN PROBLEM!

The heuristics you choose are a matter of context/values/fit/efficacy /preference

8



Avoid Modeling Gotchas

A Quick Option For Capturing Heuristic Essence

Identify Trusted Events

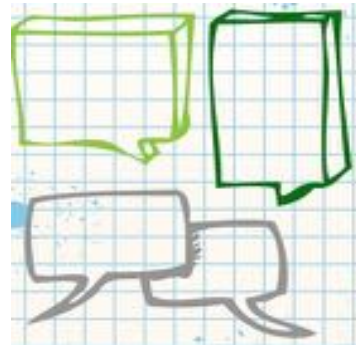
Happy Path + Expected Exception

9

Activity: Share Some Cherished Heuristics

Have a 7 minute conversation with your neighbor. Share some of your favorite modeling & design heuristics.

Write a short phrase describing each heuristic on a sticky note.



7 minute EXERCISE

10

What Values underlie Your Heuristics



Revisit that conversation you just had.

5 minute
EXERCISE

What values motivate those cherished heuristics?

Write a short phrase describing those values on sticky notes.

11



Values depend on context

- I value understandable code. So I...
 - give methods, functions, and variables meaningful names; keep code in methods short & at the same level of detail....
- As context shifts, so do my values
 - checking out a potential solution in Stack Overflow, I don't value understandability so much
 - if it will solve my problem, I rewrite the code to be understandable

12

Some of My Agile Design Values

Value evidence over speculation

Value consistency over cleverness

Design for what I know now

Learn, then adapt

Simple (uncomplicated) if possible, not simplistic



Photo courtesy of
<https://www.flickr.com/photos/notbrucelee/7113385543>

13



Values & actions depend on context

- I value tested code...
 - I also value expressive & illustrative, minimal tests....
- I often work on a design of a set of classes or cooperating functions at the same time
- So while I write tests...
 - I don't always write them before I write code
 - I often leave only tests for code that exercises lower-level functionality (& not for those lower-level functions)

14



Values & actions depend on context **and our unique design umwelt**

True Confession: A lot of my cherished design heuristics were shaped by my Smalltalk programming experience

15

PUTTING IT ALL TOGETHER:

We interleave heuristics, use several at the same time, and choose heuristics based on our individual preferences, the current context, and what we are paying attention to/noticing

16

Guiding Heuristic: Design/Code in Small Steps



- Heuristics for making progress:
 - Write a test to express what the simplest working thing can be, then make it pass.
 - Write scaffolding code.
 - It's OK to leave code ugly for a while.
 - Once working, fix the code to be closer to where you want the design to be.
- Values:
 - Tested code
 - Don't settle for ugly

17

Jason Yip's Heuristic Sequence for "Splitting" teams



Guiding Heuristic: Wait for seams to appear: e.g. clunky communications, awkward meetings...

Heuristics for taking action:

1. Nudge things apart, that is separate rhythms and events
2. Formalize the event. If done correctly this is mostly an acknowledgement of a non-event

18

TEAM AGREEMENTS


MORE POSITIVE FEEDBACK	IMMEDIATELY CLARIFY A TASK YOU DON'T UNDERSTAND
COMMIT TO ENGAGE WHEN PRESENT	ACKNOWLEDGE THE OTHER PERSON'S PERSPECTIVE
PREPARE BEFORE MEETINGS = OBJECTIVE & AGENDA =	DON'T START TALKING WHILE SOMEONE ELSE IS TALKING
ENCOURAGE A COLLEAGUE IF YOU FEEL HE/SHE NEEDS IT	POSITIVE ATTITUDE TOWARDS EACH CLIENT & CUSTOMER

Heuristics in Action: Working Agreements at Mozaic Works*

*Thanks, Alex Balboaca for sharing

19

Activity: Tell a Story and Identify Heuristics



Small group
15 minute
EXERCISE

Think of a recent design or work challenge. What actions? Did you have to back up and retry another approach? (Capture a cluster of heuristics)

20

What makes any heuristic useful?

- It fits into the way you want to work
- ... and your current design context
- It readily comes to mind
- You can adapt it as you learn



courtesy Jordan Wirfs-Brock

21

Heuristics May Conflict...



and still be useful

© Can Stock Photo / DaneeShe

22



Cucumber Team Heuristics for Driving
when Remote Mobbing

<https://cucumber.io/blog/bdd/five-roles-in-a-healthy-mob/>

23

Traditional Driving and Navigating

“The coding done by the **Driver** is simply the mechanics of getting actual code into the computer. The **Driver** is also often involved in the discussions, but her main job is to translate the ideas into code.”

“the **Navigator** is doing the thinking about the direction we want to go, and then verbally describes and discusses the next steps of what the code must do.”

–MobProgramming.org

24

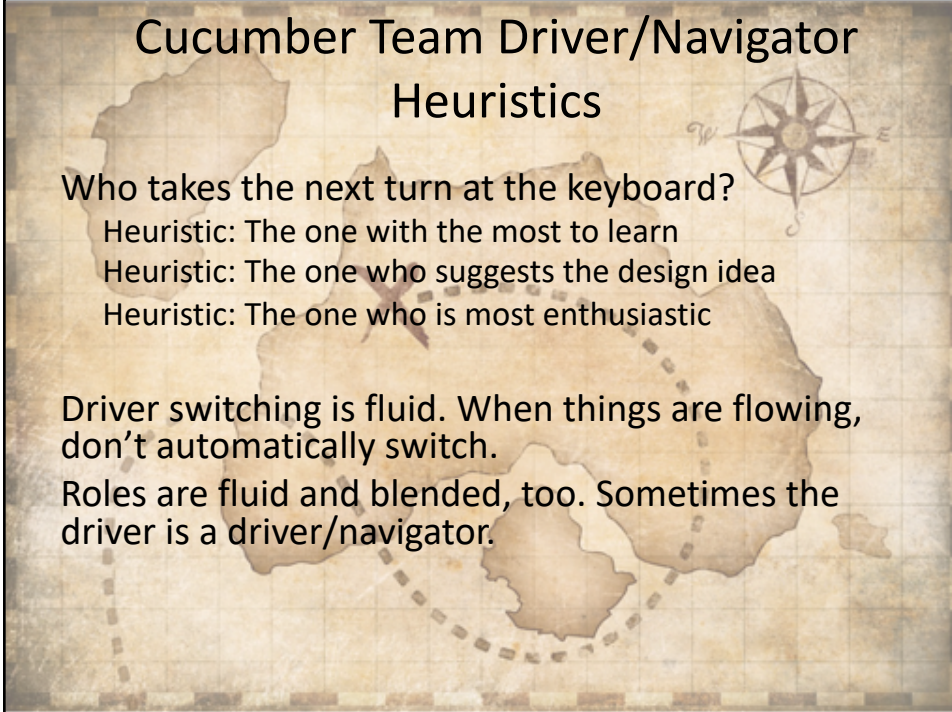
Cucumber Team Driver/Navigator Heuristics

Who takes the next turn at the keyboard?

- Heuristic: The one with the most to learn
- Heuristic: The one who suggests the design idea
- Heuristic: The one who is most enthusiastic


Driver switching is fluid. When things are flowing, don't automatically switch.

Roles are fluid and blended, too. Sometimes the driver is a driver/navigator.



25

Choose the heuristic to use from what **you** take to be the best option at the time you are required to choose.



26

Recognition-Primed Decision Making*

1. Known situation/known action

If x then do y

2. Unknown situation known/limited set of possible actions

If ??? then gather/fill in missing information and

case a: do x

case b: do y

case c: do z ...

3. Known situation/unknown action

Satisficing. Mentally simulate the consequences of taking action. Stop trying out alternative scenarios when you find an acceptable outcome



*Gary Klein, *Sources of Power: How People Make Decisions*

27

What are my “go to” heuristics?
 What makes these heuristics work?
 When might I try alternatives?
 How does the current situation impact my heuristic choices?

What else can we find by looking around?

@rebeccawb

28

A testing survey



<https://tinyurl.com/testing-poll>

29

A modeling survey



<https://tinyurl.com/Modeling-Survey>

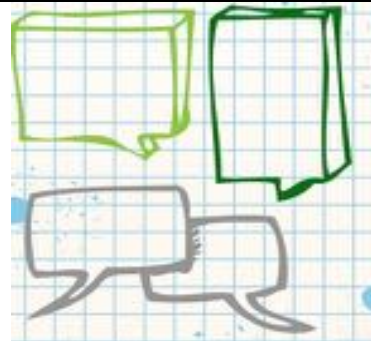
30

A documentation
survey



<https://tinyurl.com/Design-Documents>

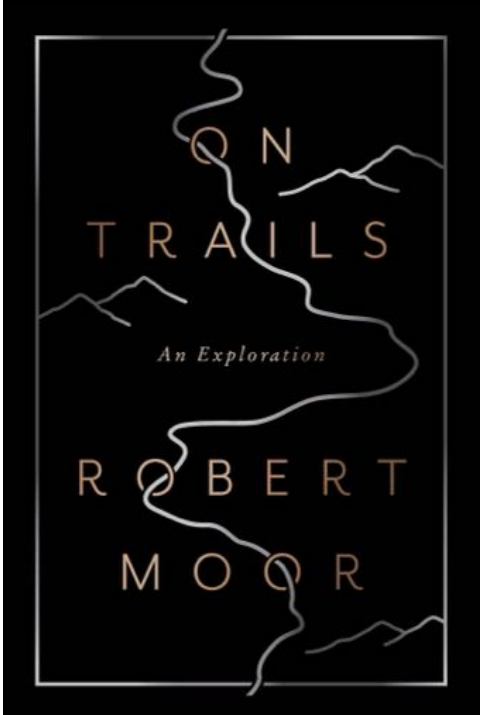
31



5 minute Discussion

Identify Some Competing
Heuristics


32



Our Heuristics Evolve

“An explorer finds a worthwhile destination; then every walker who follows that trail makes it a little better. Ant trails, game paths, ancient ways, modern hiking trails—they all continually adapt to the aims of their walkers.”

33



What do experiments look like?

34

Well-Designed Experiments

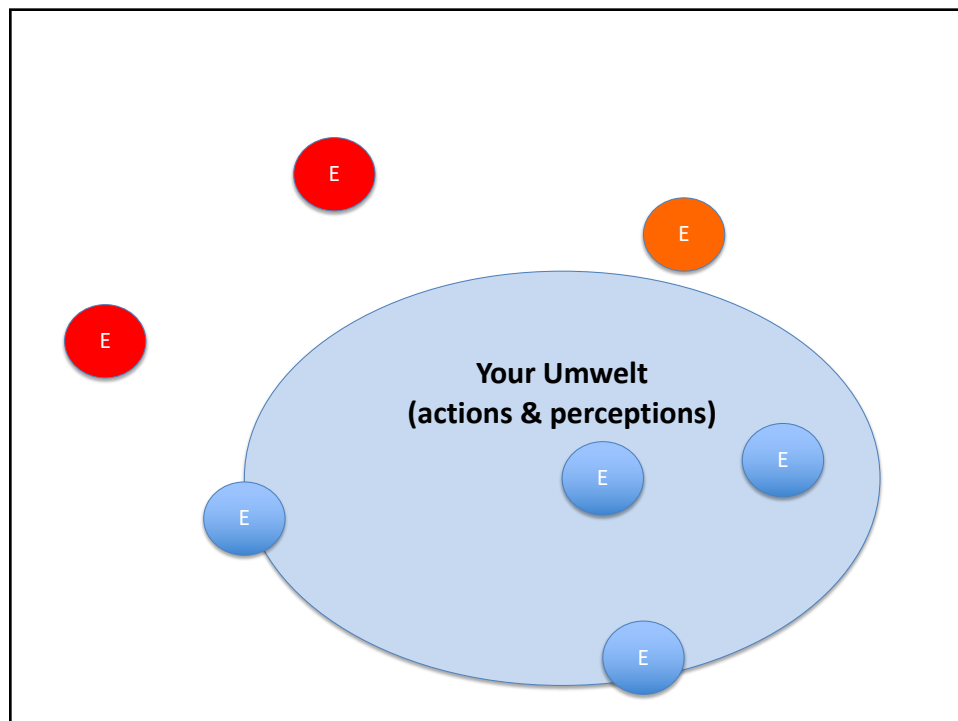
Characteristics

- Interesting
- Small
- Bounded (limited scope and duration)
- Observable results
- Results not predictable
- Reversible
- Can be run under varying conditions

Kinds

- Learning a new technique, practice, or technology
- Tinkering with improvements (nudges, living documentation, code reviews,...)
- Comparing competing heuristics for design and modeling and architecture

35



36

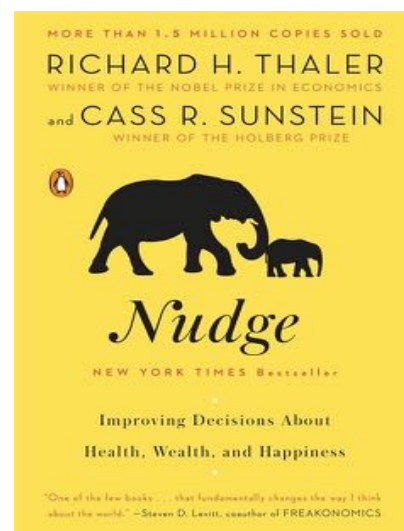
Steps: Experiment Design

- What actions you will take? What will you try?
- What are the anticipated results?
- What are some possible side-effects?
- What do you hope to learn?
- What next?

37

“A nudge is any aspect of the choice architecture that alters people’s behavior in a predictable way without forbidding any options or significantly changing their economic incentives. To count as a mere nudge, the intervention must be easy and cheap to avoid. Nudges are not mandates. Putting fruit at eye level counts as a nudge.”—Richard Thaler and Cass Sunstein, *Nudge*

NUDGES



38

10 kinds of nudges

- default rules
- simplification
- use of social norms
- increases in ease or convenience
- disclosure
- warnings, graphics or otherwise
- precommitment strategies
- reminders
- eliciting implementation intentions
- informing people of the nature and consequences of past choices

Nudging a Very Short Guide:
<https://dash.harvard.edu/handle/1/16205305>

39

Almost There

CONTRASTING NUDGES

“The Garmin watch vibrates and displays the move bar with the word move hourly if you've been stationary. When you start moving the move bar will decrease and the watch vibrates again when it's cleared. What exactly are you looking for and what do you mean by using the feature more effectively? I'm not sure how it could be anymore effective. There are watch faces that continuously display the level of the move bar. Is that something you're looking for?”

Move bar cleared!

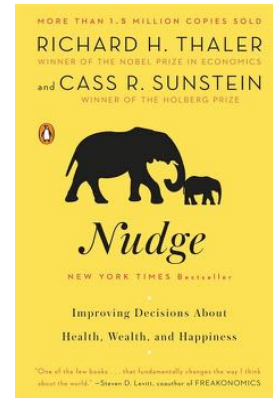
40

Nudges to Improve Design & Architecture

"..people will need nudges for decisions that are difficult and rare, for which they do not get prompt feedback, and for which they have trouble translating aspects of the situation into terms they can understand."

Some possible nudges:

- checkpoints (instead of Big Phase Gate staged processes or free-for-all-no-checks-or-balances)
- default support for preferred configurations
- code analysis at check in
- GitHub Product Release templates



41

GUIDELINES ARE HEURISTICS, TOO



Land Ice Working Group Developers' Guidelines

<https://tinyurl.com/Developer-Guidelines>

HIPPO Data Set Documentation Guidelines

<https://tinyurl.com/Data-Set-Doc-Guidelines>

Guidelines on Communicating Uncertainty in Forecasts

<https://tinyurl.com/Documenting-Uncertainty>

they cause us to take action

42




20 minute
Exercise

DESIGN AN EXPERIMENT

to Expand Your Heuristics

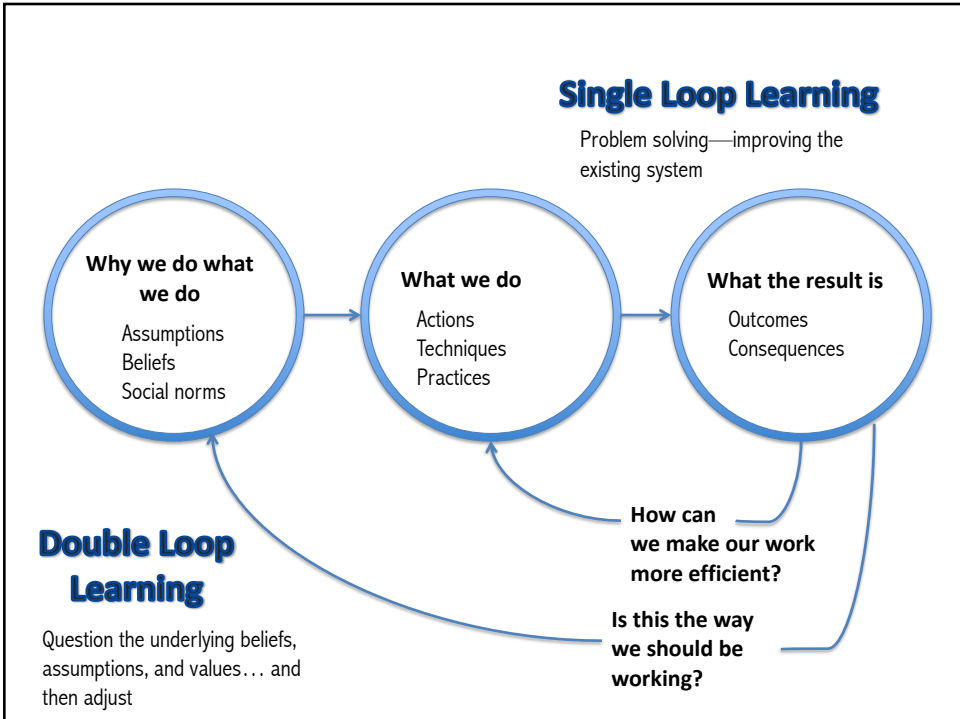
43



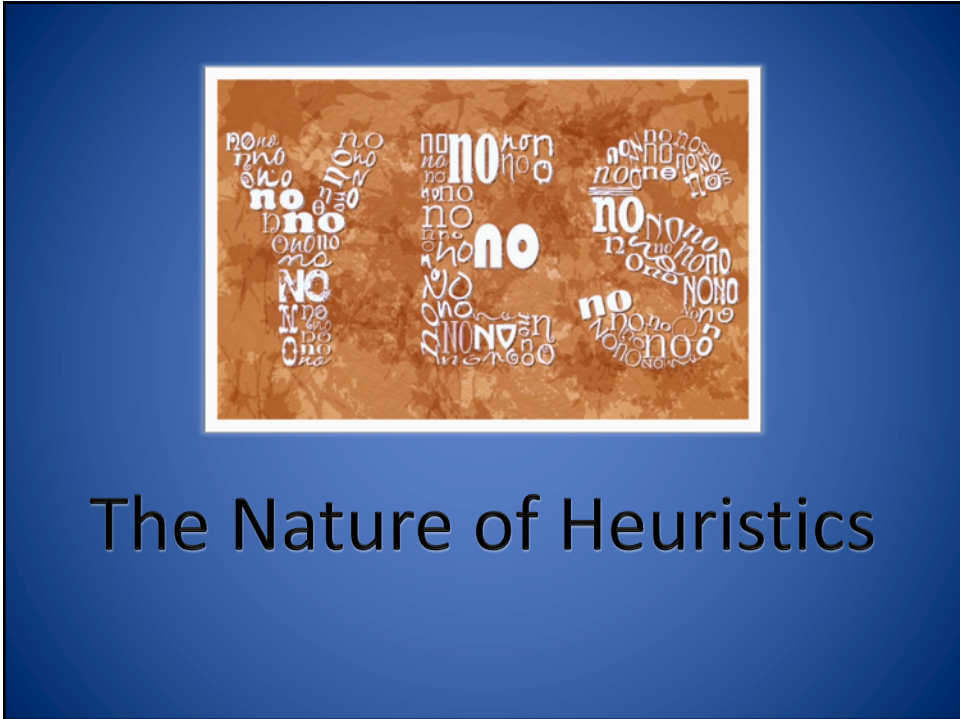
Experiment Design

- What actions you will take? What will you try?
- What are the anticipated results?
- What are some possible side-effects?
- What do you hope to learn?
- What next?

44



45



46

Not all Heuristics are Alike

- Some are specific to a particular context
- Some work in many different contexts
- Some require us to rethink or readjust our design values
- Some require significant practice to master

47

What makes any heuristic useful over the long term?

- It repeatedly solves your problems.
- You understand its limitations. If it doesn't fit, you don't use it. You try something else.
- With enough knowledge about how and why it works, you can adapt it to new situations.



courtesy Jordan Wirfs-Brock

48

Heuristics

- We all hold a lot of heuristics in our heads
- We often don't clearly articulate, share, or communicate our heuristics (we tried to in this workshop)
- There is a subjectivity to our heuristics; they hold our history
- We each hold different heuristics for deciding and acting in similar contexts.
- The goal is not to get everyone aligned on the same set. Evolvability and sustainability of a complex system benefit from multiple ways of seeing and interacting with its design and architecture

49

Further reading: Heuristics blog posts

<https://chelseatroty.com/2023/03/22/testing-a-heuristic-hunting-conversation-with-rebecca-wirfs-brock/>
[https:// wirfs-brock.com/blog/2023/04/19/testing-testing-our-heuristics/](https://wirfs-brock.com/blog/2023/04/19/testing-testing-our-heuristics/)
<https:// wirfs-brock.com/blog/2023/05/02/our-heuristics-are-shaped-through-experience/>
<https://wirfs-brock.com/blog/2023/05/04/getting-out-of-your-ruts/>
<https://wirfs-brock.com/blog/2021/12/17/too-much-salt/>
<https:// wirfs-brock.com/blog/2019/03/20/growing-your-personal-design-heuristics>
<https://wirfs-brock.com/blog/2019/04/04/what-do-typical-design-heuristics-look>
<https:// wirfs-brock.com/blog/2019/04/12/writing>
<https://wirfs-brock.com/blog/2019/04/19/nothing-ever-goes-exactly-by-the-book>
<https:// wirfs-brock.com/blog/2019/04/25/what-we-say-versus-what-we-do>



Thank you!
rebecca@wirfs-brock.com

www.wirfs-brock.com

50

Further reading: Heuristics and pattern essays

“Observations on Growing a Software Design Umwelt”
by Rebecca Wirfs-Brock, PLoP 2022
[https://wirfs-brock.com/PDFs/
Observations on growing a software design umwelt.pdf](https://wirfs-brock.com/PDFs/Observations%20on%20growing%20a%20software%20design%20umwelt.pdf)

“Elephants, Patterns, and Heuristics”
by Rebecca Wirfs-Brock and Christian Kohls, PLoP 2019,
[https://wirfs-brock.com/PDFs/ Elephants, Patterns, and Heuristics.pdf](https://wirfs-brock.com/PDFs/Elephants,%20Patterns,%20and%20Heuristics.pdf)

“Are Software Patterns Simply a Handy Way to Package Heuristics?”
by Rebecca Wirfs-Brock, PLoP 2017
[https://wirfs-brock.com/PDFs/
AreSoftwarePatternsSimplyaHandyWaytoPackageDesignHeuristics.pdf](https://wirfs-brock.com/PDFs/AreSoftwarePatternsSimplyaHandyWaytoPackageDesignHeuristics.pdf)

“Traces, tracks, trails, and paths:
An Exploration into How We Approach Software Design”
by Rebecca Wirfs-Brock, PLoP 2018
[https://wirfs-brock.com/PDFs/
Traces, tracks, trails, and paths- An Exploration of How We Approach Software Design.pdf](https://wirfs-brock.com/PDFs/Traces,%20tracks,%20trails,%20and%20paths-%20An%20Exploration%20of%20How%20We%20Approach%20Software%20Design.pdf)



Thank you!
rebecca@wirfs-brock.com

www.wirfs-brock.com