

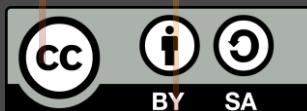
BASIC TOOLMAKING: THE ROAD TO EXTENSIBILITY

Evgenij Smirnov

@cj_berlin

April 8-11, 2024

1





PURE STORAGE®



ScriptRunner®
The #1 for PowerShell Management

PDQ DataOn®



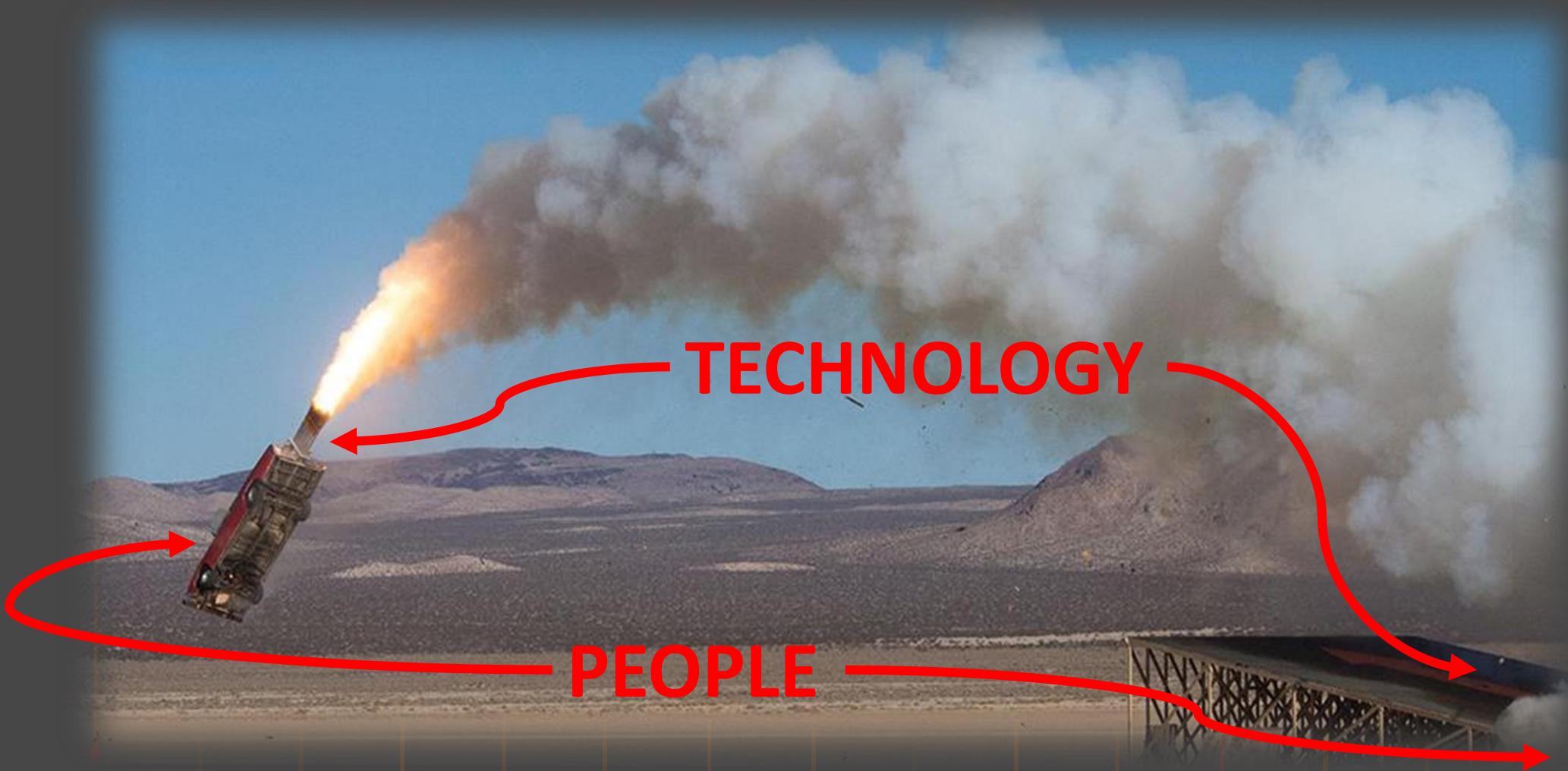
about_Speaker

```
PS C:\> Get-SpeakerInfo

Name      : Evgenij Smirnov
Twitter   : @cj_berlin
BlogSite  : https://it-pro-berlin.de
Location   : Berlin
YearOfBirth : 1972
JobTitle   : Senior Solutions Architect
WorksAt    : Semperis
MVPSince   : 2020
UserGroups : {WSUG, EXUSG, PSUGB}
SpokeAt    : {PSConfEU, CIMLingen, PSConfEU, PSDAY.UK...}
CertifiedIn : {Microsoft, VMware, Quest, ITIL}
```



What is this about?



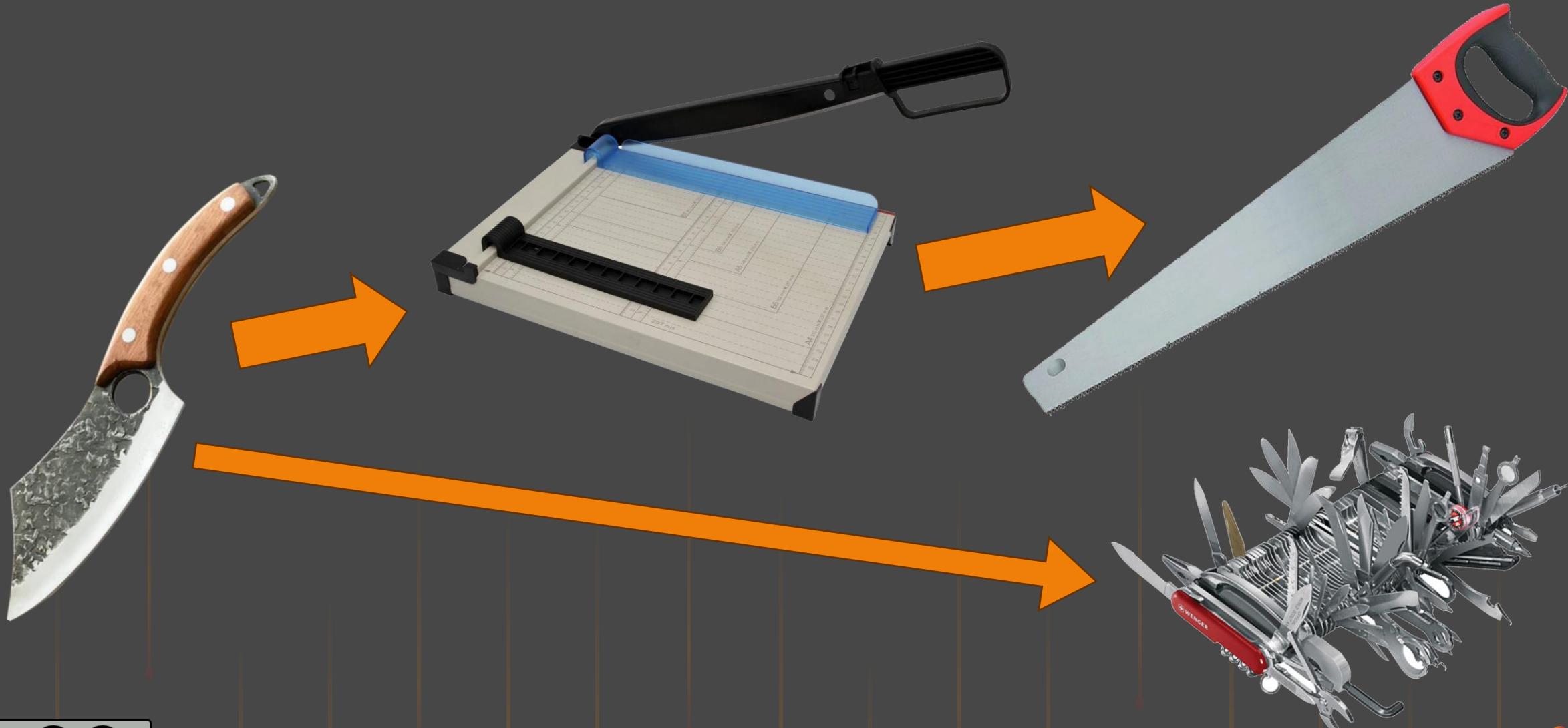
So you built a tool...



April 8-11, 2024

5

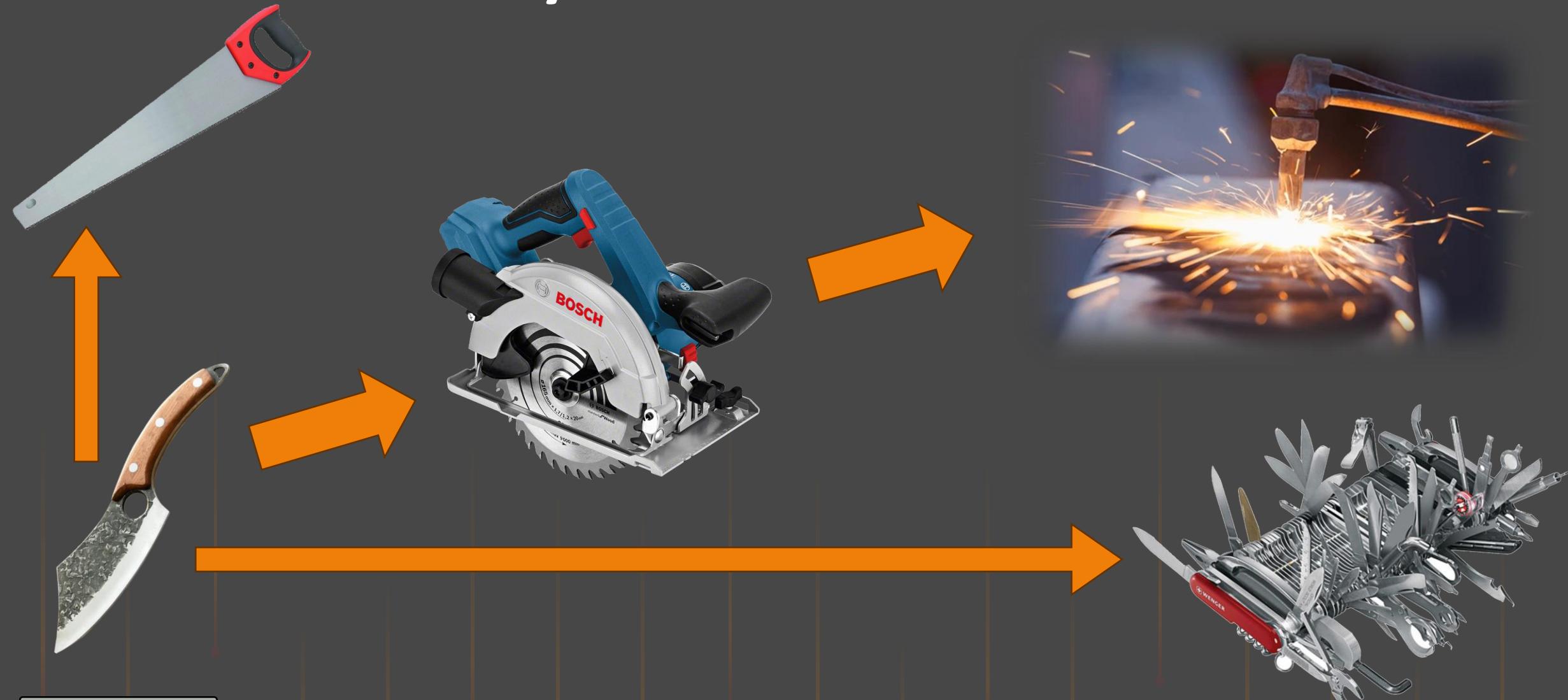
So you built a tool...



April 8-11, 2024

6

So you built a tool...



April 8-11, 2024

7

How do we deal with this?

(and still preserve our sanity)

April 8-11, 2024

8



stating the obvious...

„Weeks of coding can save you hours of planning“

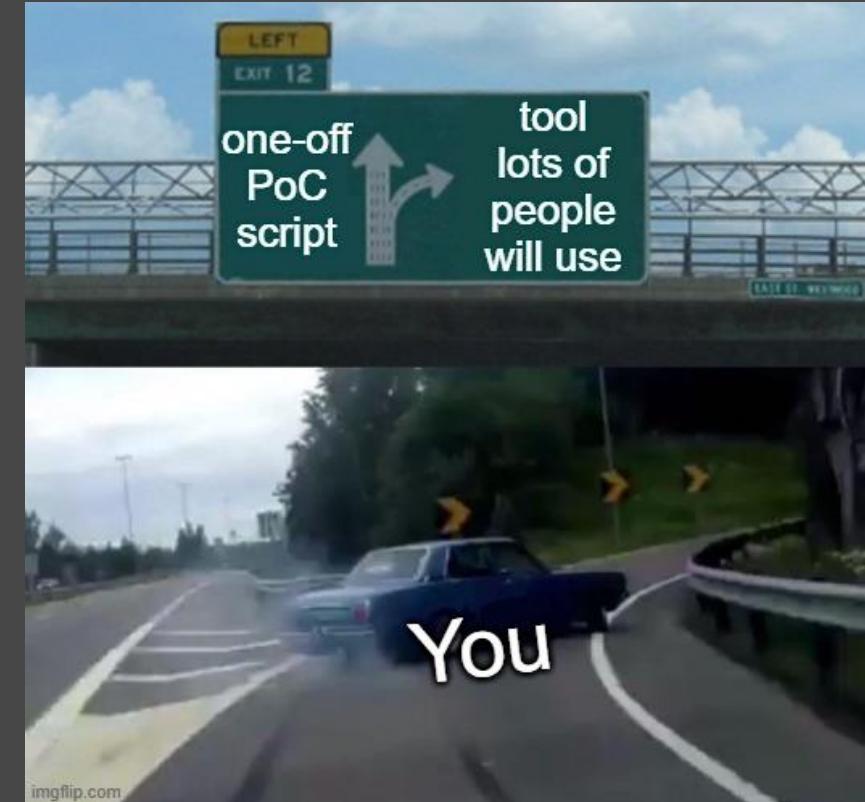
- *Frank Sons, around 2018*

Unfortunately, practical toolmaking does not work like this.

Assume Responsibility

- Develop what you promised – but also...
- ...test for changing external conditions
- ...collect feedback and act accordingly
- ...support where and when needed
- ...find contributors – **your health matters!**

00-00



Before you publish 0.0.1

Because nothing kills user acceptance
like breaking changes

Packaging & Shipping

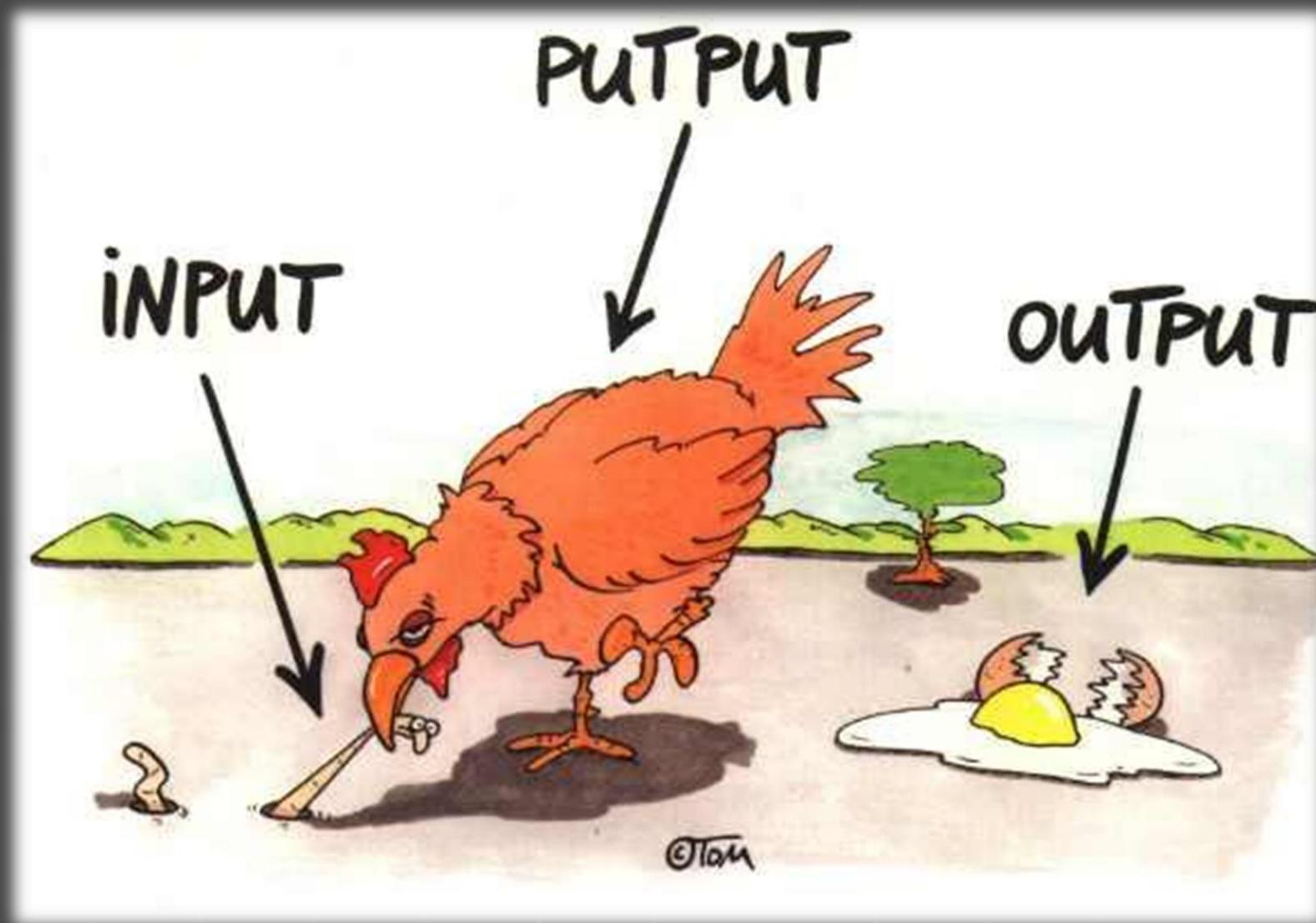
- Single script, a suite of scripts or a module?
- Module
 - has to be imported → target group may not know how
- Script
 - is best at doing one thing, maybe in different modes
- Suite of scripts
 - scripts have to be packaged together and named consistently



Package naming

00-01

- Names are but sound and smoke... *not*
- Modules: Don't include specifics, those are for function names!
- Script: Including actions helps, but do not include
 - Targets
 - Protocols



Expecting Input

- Every script (and public function) should exhibit SOME non-red behavior when called without parameters
 - Because people are people.
 - Parameter sets may take you down a rabbit hole of dynamic params
 - Windows GUI for input is not always very extensible
 - Pode.Web can be a great portable alternative (*if you can include it*)
 - TUI can be a great alternative for the default behavior



Output: What they get

- Output files
 - Keep default locations once you start using them
- Log files
 - Keep file locations and naming conventions
 - Log naming can absolutely be function-dependent (**remember to restart**)
 - Log the functionality being invoked – either at the start of an invocation or as line prefix

Output: What they see

- **Keep your initial output mode as an option!**
- Console output: prepare for scale
 - Use `Write-Progress` (even if it kills speed in Windows PowerShell)
- WPF or WinForms GUI can be extensible but lots of work
- If nice output and interactivity both have to be extensible, consider Pode.Web
- HTML Report as output can solve many problems
 - Trying to auto-open that report will CAUSE many problems ;-)

Under The Hood

What your users don't need to know
but will appreciate



April 8-11, 2024

18

Having Dependencies

- If your script or module has dependencies, do **enforce them**
 - **PLEASE DO NOT ALTER THE TARGET SYSTEM WITHOUT CONSENT!**
- **#requires** prevents execution but has limited capabilities
- Early breakout can check everything but has to be really early
 - can affect initialization
 - must roll back completely
- Late enforcement can help by very complex scripts or modules

Coding Techniques

02-00

- Functions, functions everywhere...
- Use master functions to split business logic from functionality
- Fully qualified cmdlet names where warranted (e.g. Hyper-V + VMware)
- Dynamic parameters help protect end users from mistakes
- Different approaches to state preservation in different systems



April 8-11, 2024

20

Error handling

- New functionality potentially makes the code less stable
- You can't always ask for logs, but being made aware of unhandled exceptions is **critical**
- Provide a global „Red Alert“ functionality using trap{}
 - Works differently in modules!!!!
 - This is only valuable if the end users actually send you the file ☺

03-00



Development Process

- Split the functionality into functions
 - makes the logic in the master script/function easier to understand
- Split the functions out into separate files and use pipelines
 - makes both editing and reusing easier, also change tracking and collab
- Keep non-master functions reusable
 - Specifically, try to avoid script-scope variables
- Document and test

10 Questions

to take home from this session

- Am I prepared to assume responsibility → **Time / Collaborators**
- Who are the target audience? → **Packaging / Shipping**
- How do I know they really want X Y Z? → **Collecting feedback**
- Where is the red line, functionality-wise? → **Start a new project**
- What are the most important use cases? → **Parameter handling**

10 Questions

to take home from this session

- What is the expected UX? → **Output / Reports / Interactivity**
- What are the dependencies today and tomorrow? → **Handling**
- How will I inform users of new extensions? → **Online/Inline help**
- How to prioritize feature requests? → **Impact on usage vs. effort**
- What if I can't maintain this anymore? → **Documentation**

THANK YOU

Please review this session



Session Review

April 8-11, 2024