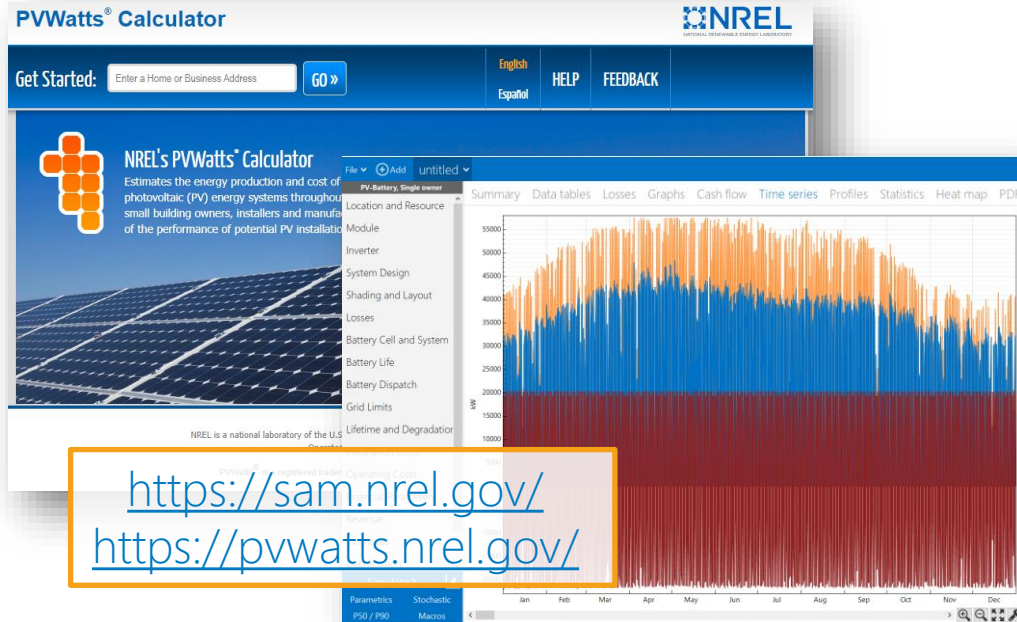


Successes, Lessons Learned, and Ongoing Challenges from the Open-Source Release of the System Advisor Model

Brian Mirletz
SETO-funded Open-Source Software Workshop
10/12/2022

System Advisor Model (SAM) & PVWatts

Free software that enable detailed performance and financial analysis for renewable energy systems



- ✓ Desktop application
- ✓ PVWatts web tool & API
- ✓ Software development kit
- ✓ PySAM Python package
- ✓ Open-source code
- ✓ Extensive documentation
- ✓ User support



Technologies

- Photovoltaic
- Energy storage
 - Electric battery
 - Electric thermal storage
- Concentrating solar power
- Industrial process heat
- Marine energy
- Wind power
- Fuel cell
- Geothermal power
- Solar water heating
- Biomass combustion
- Generic system

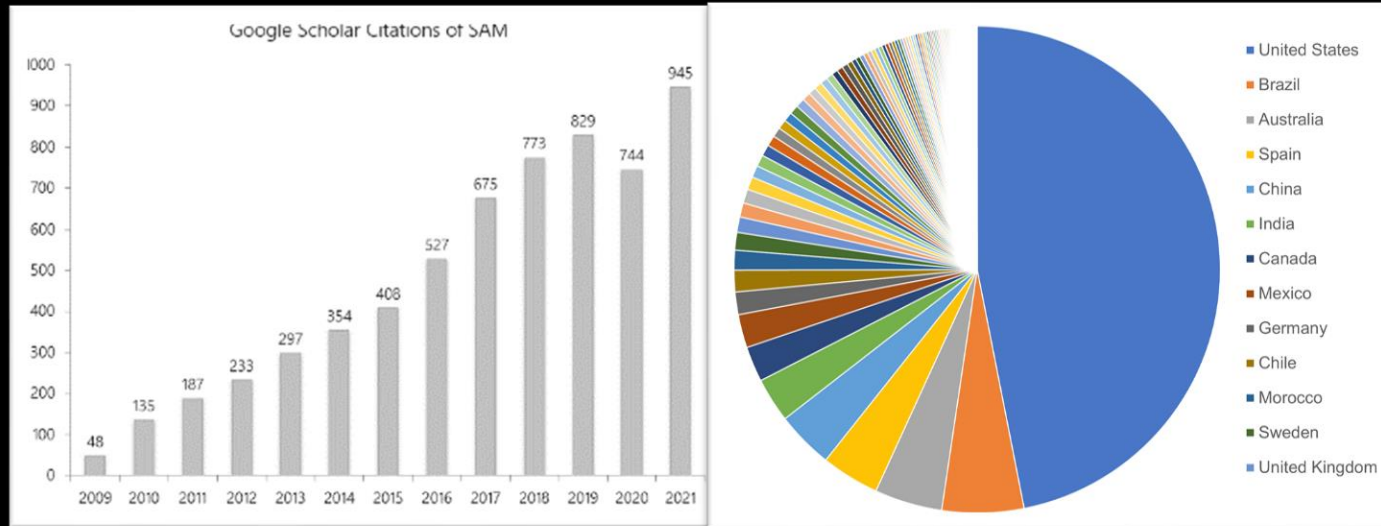
Financial Models

- Power purchase agreements
 - Single owner
 - Partnership flips
 - Sale leaseback
- Residential
- Commercial
- Third party ownership
- Merchant plant
- Community solar
- Simple LCOE calculator

SAM Users

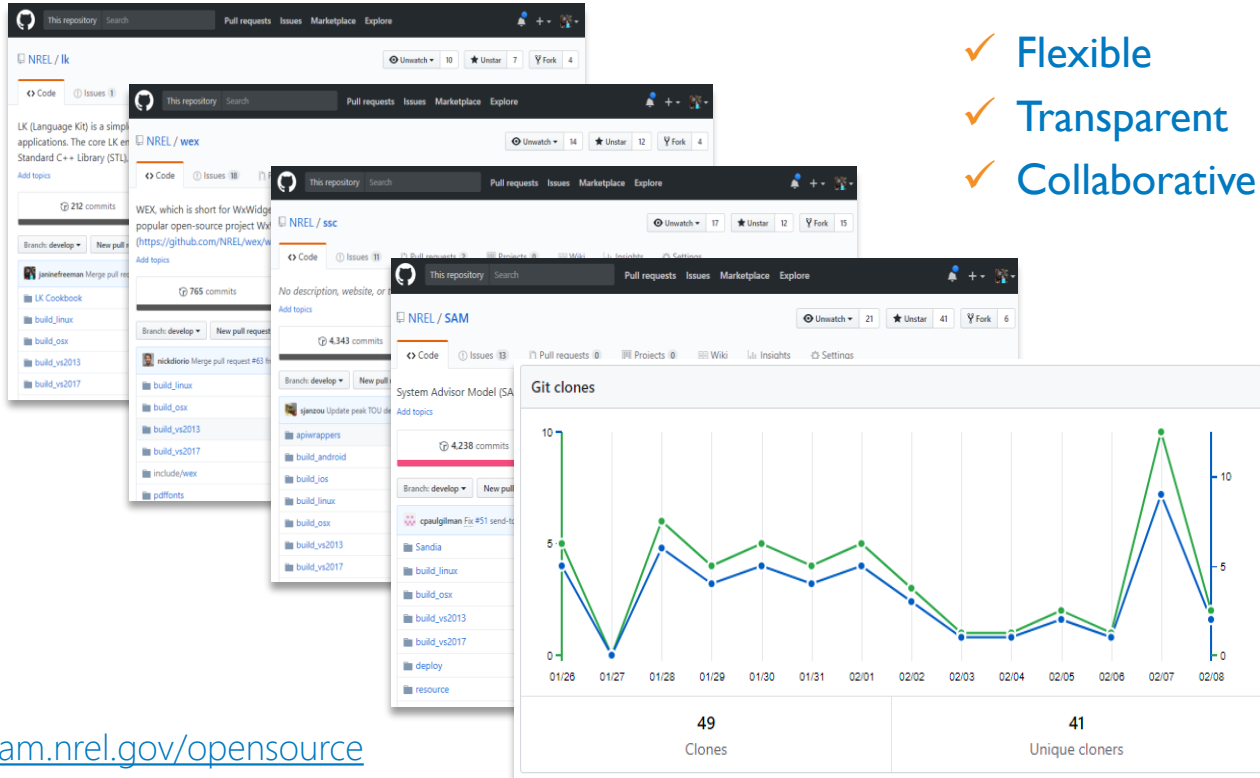
SAM is started **once every 2 minutes**
PVWatts receives over **17.5 million hits per month**
Over **150,000** users in 190+ countries
120+ webinars with **over 280,000 views**

Users include Sunrun, Enphase, AEP, Southern Company, EPRI, & more



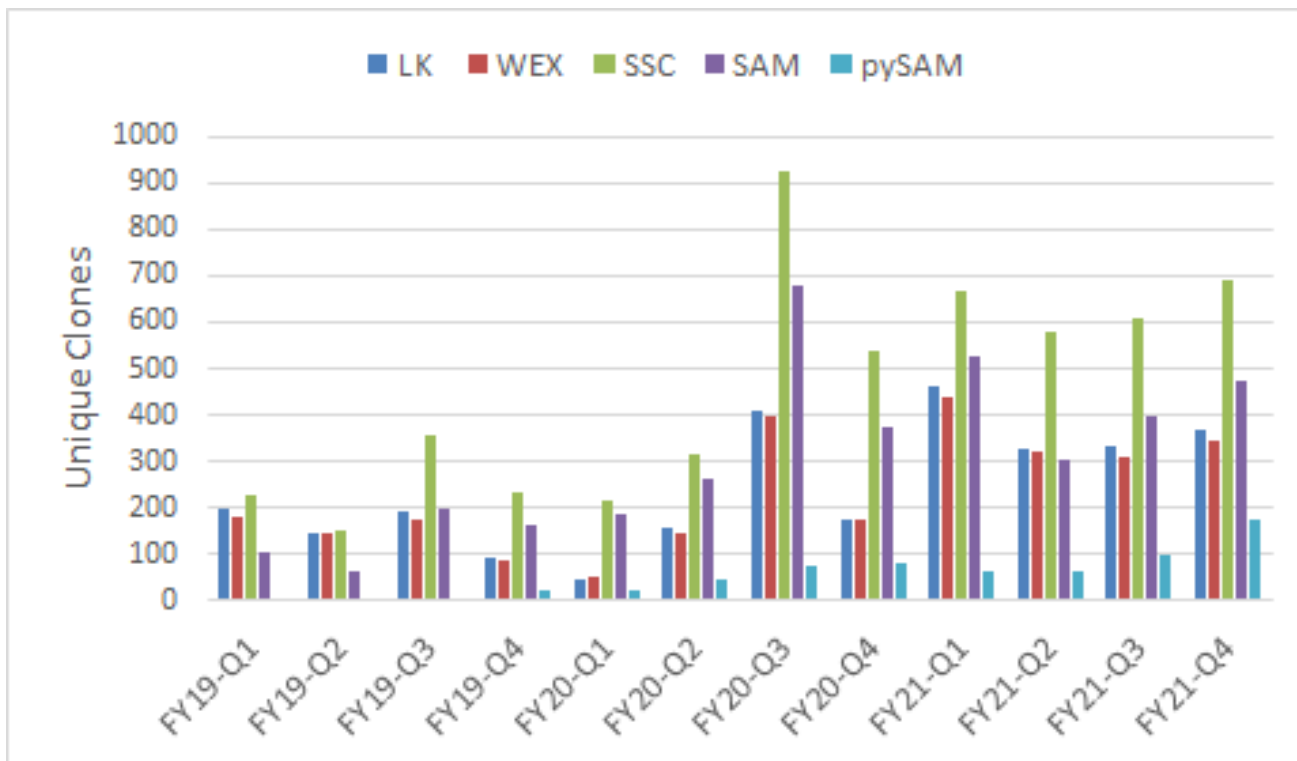
Open-Source Code

- ✓ Flexible
- ✓ Transparent
- ✓ Collaborative



<http://sam.nrel.gov/opensource>

Open-Source Usage



History

Developed by

- Department of Energy
- National Renewable Energy Laboratory
- Sandia National Laboratories

Originally launched closed-source in 2004

- Model different renewable energy projects in a single platform
- Facilitate technology comparison by handling performance, costs and financing consistently across technologies



Motivation for Open-Source

Open-source Launch in 2017. Provides:

Transparency

- Look at the underlying code of models

Flexibility

- Can tweak models to represent a new or unusual configuration

Standards

- Provides standards for all models, including closed-source

Collaboration

- New technology models that might not yet have a commercial market
- Encourages collaboration amongst taxpayer funded projects

Licensing

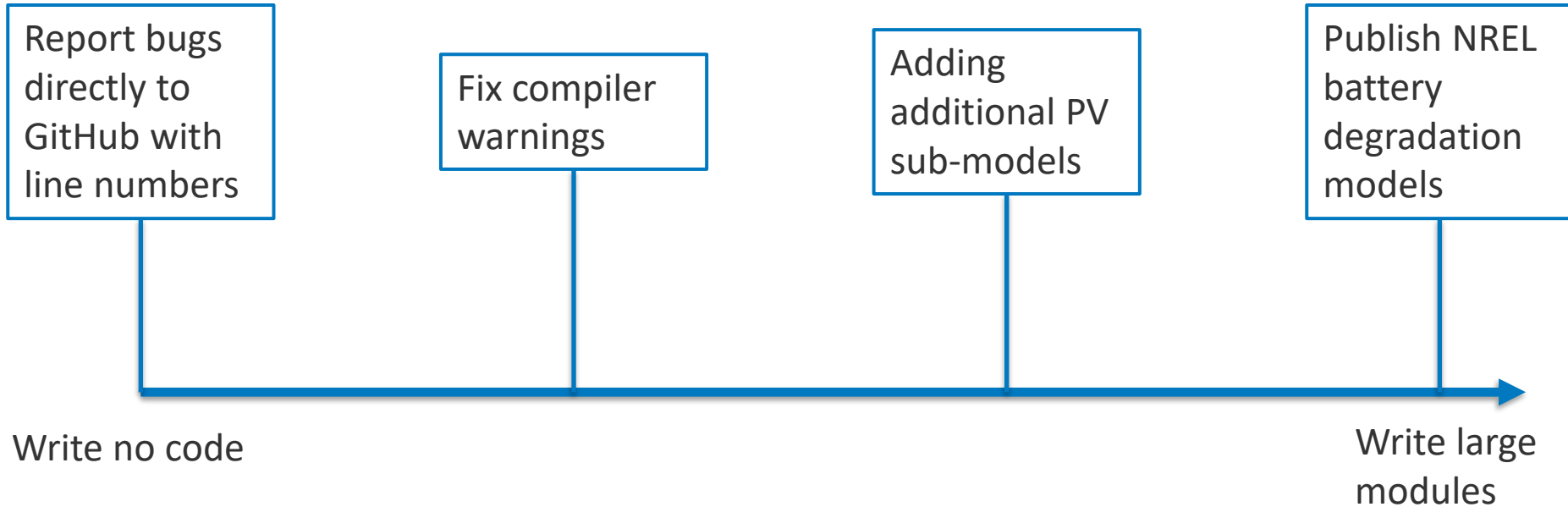
Entity	2017-2019 License	Current License
Commercial Entities	MIT-type license with no sharing restrictions	BSD3-clause (no sharing restrictions)
Research and Non-profit Entities	GPLv3 like license with sharing required	BSD3-clause (no sharing restrictions)

Lesson learned: standard licenses make it easier for contributors

Open-Source Contributors

- Other NREL teams
- Sandia National Laboratories
- Cypress Creek Renewables
- Envision Digital
- OpenInvest
- Passivenous Consulting
- Southern Company

Types of Contributions



Large Successful Contributions

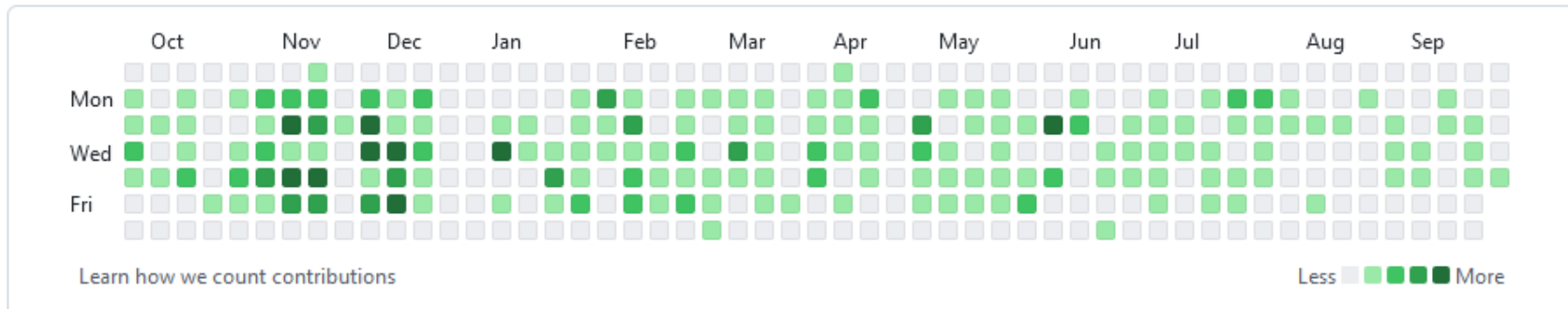
- Mermoud/Lejeune module model
 - Included unit tests, released as an SDK option
 - Received subsequent open-source contributions with extensions and fixes
- Slope-aware backtracking algorithm
 - Able to keep the original author aware of updates
- Chemistry specific battery degradation algorithms

Reasons to Contribute

- Prestige
 - Contributions are public
- Solving business needs
- Unofficial “stamp of approval”
 - Able to say changes are in the official NREL SAM repository

931 contributions in the last year

Contribution settings ▾



Challenges

- Language choice
 - C++ is fast, but smaller pool of developers
- Desktop GUI increases user base, but adds another barrier to contributions
 - Core team has needed to develop GUI interfaces for some contributed features
- Keeping some code private (usage tracking, API keys) adds development overhead

Contributor Outreach

- Outreach is different when the primary interfaces is a GUI (vs code)
- Use in university courses provides opportunities to get programmers into renewable energy
- “Volunteer” contributors are rare
 - Contributors are usually affiliated with the energy field professionally
- How do we get companies to view using and contributing to open-source tools as an advantage?

Thank you! Questions?

Janine (Freeman) Keith – project lead, photovoltaic and wind models

Nate Blair – emeritus lead, financials, costs, systems

Darice Guittet – software development, battery models

Brian Mirlletz – software development, costs, battery models

Matt Prilliman – photovoltaic, geothermal, and marine energy models

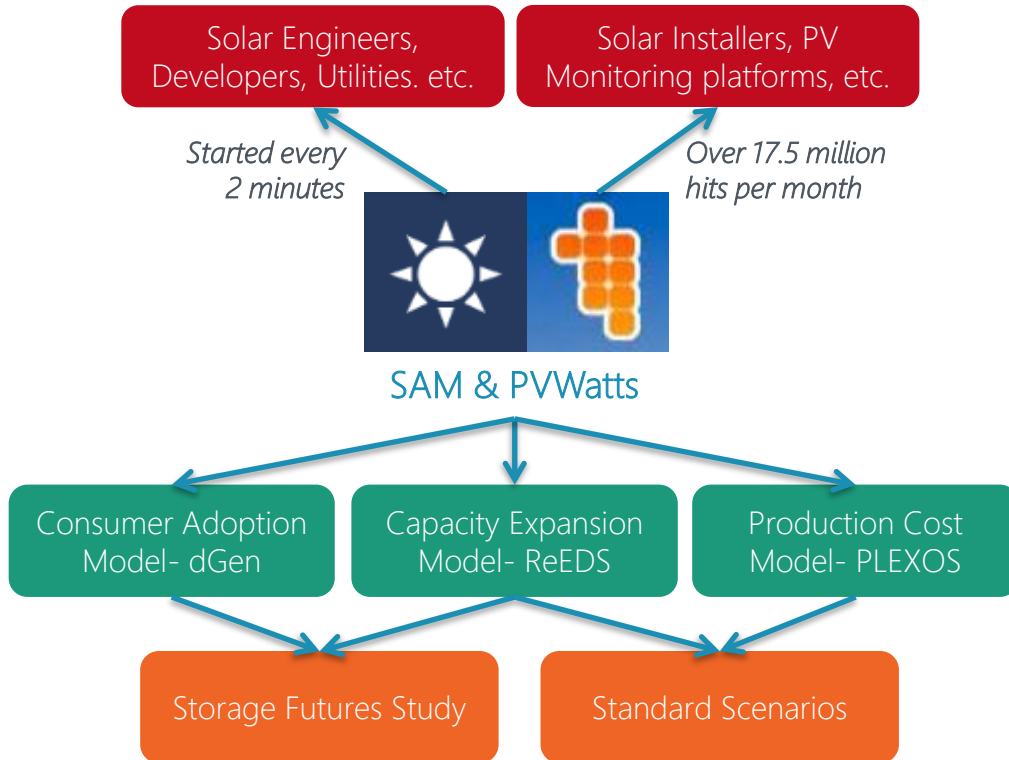
Steve Janzou – programming, utility rates, financials (subcontractor)

Paul Gilman – user support and documentation (subcontractor)

Ty Neises – concentrating solar power models

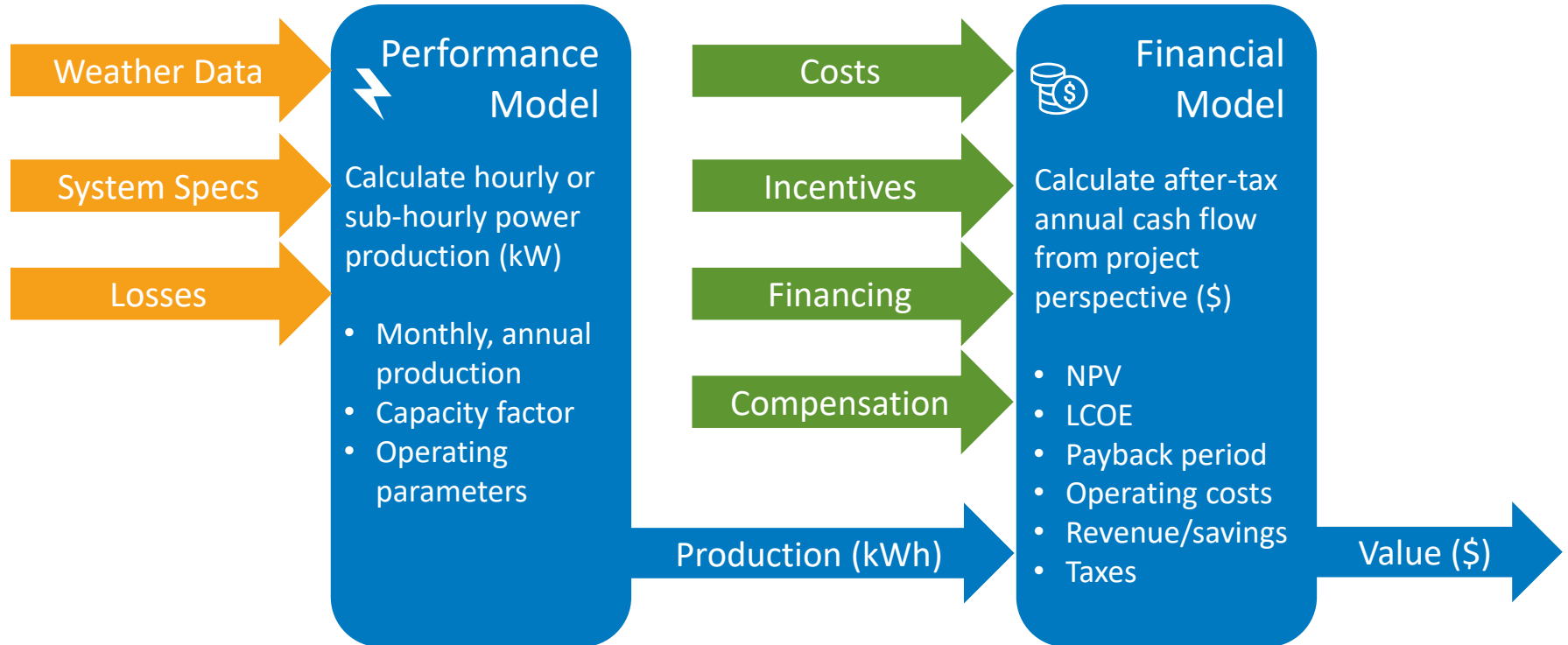
Matt Boyd – concentrating solar power models

How SAM Fits in at NREL and Externally



- ✓ Grid integration studies
- ✓ Renewable energy futures
- ✓ LCOE of breakthrough technologies
- ✓ Policy and utility rate design
- ✓ Technical potential studies
- ✓ Commercial applications (e.g. Southern Company, AEP, Sunrun)

Model Structure



How can you access SAM models?

- Desktop Application
- Advanced Analysis Features
 - Parametric
 - Stochastic
 - P50/P90
- Built-in Scripting Language
- Macros
- Software Development Kit (SDK)
 - Python (PySAM package)
 - C/C++
 - Matlab
 - PHP
 - C#
 - Java
 - VBA
 - iOS / Android
- Web Services API (PVWatts Only)
- **Open-source SAM code**

Built in Scripting Language and Macros

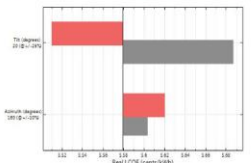
```
21 <li> Press 'Run macro' to perform the simulations and create the tornado chart.
22 </li> You can right click on the plot window that pops up to export the data or figure.
23 </ol>
24 @*/
25 // Macro user interface widgets
26 // @ name=inputs;type=inputs;Label=Input variables\nto consider;;meta=true;prompt=Specify
27 // @ name=output;type=svoutput;Label=Output metric:
28 // @ name=percent;type=number;Label=% adjustment;;value=10
29 // @ show_save_load_buttons=true
30
31
32 if ( typeof(macro) == 'unknown' ) {
33     msgbox('This macro must be run from within
34     exit;
35 }
36
37 outvar = macro.output;
38 percent = macro.percent;
39 vars = macro.inputs;
40 if ( #vars == 0 )
41 {
42     msgbox('No input variables selected. ');
43     exit;
44 }
45
46 if ( outvar == '' ) {
47     msgbox('Please choose an output variable to
48     exit;
49 }
50
51 vi = varinfo(outvar);
52 outlabel = vi.label;
53 if ( strlen(vi.units) > 0 ) outlabel = outlabel
54
```

Append Snow Data
Subarray Layout Optimization
System Sizing
Download Electric Load
Value of RE System
Combine Cases
Create a Tornado Chart
Download Weather Files
Siting Considerations
Solar Resource File Checker
Solar Resource File Converter
Solar Resource Interpolation

Create a Tornado Chart

Tornado charts can be a helpful way to visualize sensitivities of a model to various inputs. Creating a tornado chart involves running several simulations decreased and increased independently to see how much a particular output metric changes.

This macro creates a tornado chart like this one based on input ranges you specify:



Instructions:

- Using the interface at the right, select one or more input variables to consider.
- Select an output metric to plot on the tornado chart.
- Specify the percentage change (decrease and increase) to apply to each input variable.
 - A custom percentage decrease and increase, such as "10%" or "23%".
 - A custom absolute change, such as "5". If the base case input has a value of 30, values of 25 and 35 will be used.
 - A custom absolute changes in both directions, such as "4, 7". If the base case input has a value of 30, values used will be 26 and 37.
- Press 'Run macro' to perform the simulations and create the tornado chart.
- You can right click on the plot window that pops up to export the data or figure.

Flexible, lightweight scripting language built in to the SAM desktop tool, allowing users to quickly run custom analyses and read/write to other files