Observations on open data and open source

Michael Tiemann
VP, Open Source Affairs
Red Hat

#1: Define the problem

Earth science doesn't have a data problem
Earth science doesn't have a technology prob.
Earth science has a *community* problem

(nb Design Thinking approach lifted from http://opensource.org/node/158)

#2: Research

Entrez cross-references 38 databases which, together, handle 1.8B queries for 1M researchers annually, delivering 4TB of data per day. The databases range from molecular entities to nucleotides and proteins to genomes and genetic sequences to organisms and populations. It essentially represents the scientific output of $28B/year.

PLoS publishes nearly 10,000 articles a year at 1/10th the cost of Elsevier (which publishes 250,000 per year).

The Open Source community manages >= 1B SLOC.

See http://opensource.com/health/12/4/power-of-1-how-open-innovation-changed-global-health

#2 Research (cont)

rOpenSci: Access to scientific data is rapidly emerging as a central theme in the future of research. [... N]ew requirements for data management plans from NSF, data deposition requirements of prominent journals, and the emergence of well-developed repositories like GenBANK, Dryad, TreeBASE, DataONE, (KNB, NBII, GBIF) have opened a wealth of potential.

The Global Biodiversity Information Facility (GBIF) and Pensoft Publishers, through their journals ZooKeys, PhytoKeys, and BioRisk, have finalized a pilot workflow that enables the publication of biodiversity and ecology data as stand-alone, peer-reviewed, data papers generated automatically in the form of XML-based manuscripts from metadata decscriptions. [...] The Global Unique Identifier (GUID) of the published data set is cross-linked to the Digital Object Identifier (DOI) of the data paper, to allow further possibilities for data usage and data citation metrics. In addition to providing credit to scientists and revealing data resources, data papers will offer a role for academic and scholarly publishers to make data publicly and permanently available to all.

#3: Ideate (Generate ideas)

How can you address the *community's* problems?

How can you help them solve your problem by better solving their own problems?

Hint: lower cost to develop, validate, access, analyze, and archive data while *at the same time* strengthening the scientific results

(Others have done this!)

#4: Prototype

Pick something easy and prove you haven't made it impossible

Pick something really hard and prove you have made it easier

Make people not only want to participate, but *demand* to participate

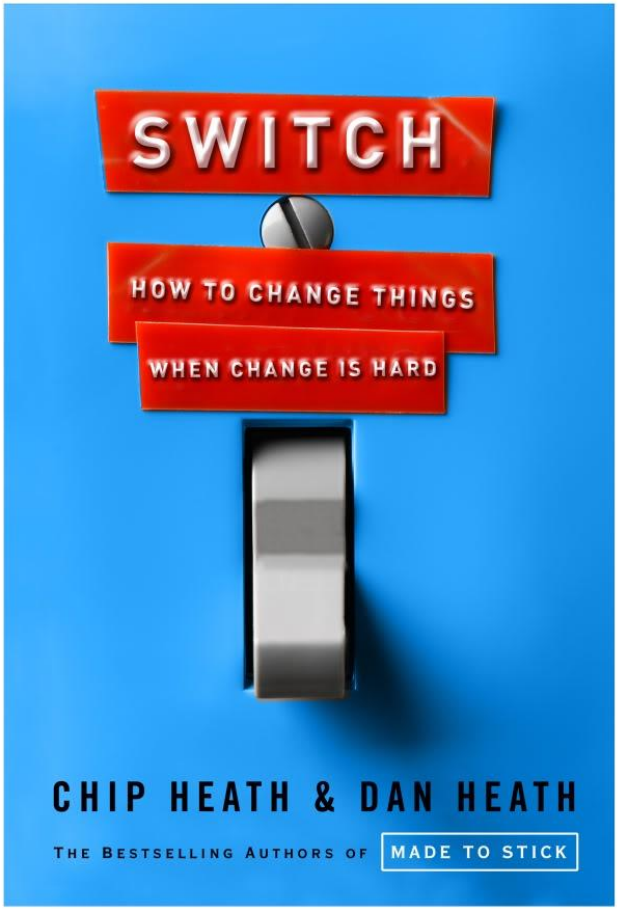
#5: Choose

Make policy and technology choices based on experiences and a strong, shared vision

"Rough consensus and running code"
(attributed to Linus Torvalds)

#6: Implement

Just do it!



#7: Learn (and do better next time)

Follow through by finding the weakest elements and fixing them.

"Fail faster so you can succeed sooner"

Fedora Project is famous for "blowing up" whatever is the biggest bottleneck, even when it's the most important component in the system! (Fast iteration cycles make this acceptable.)