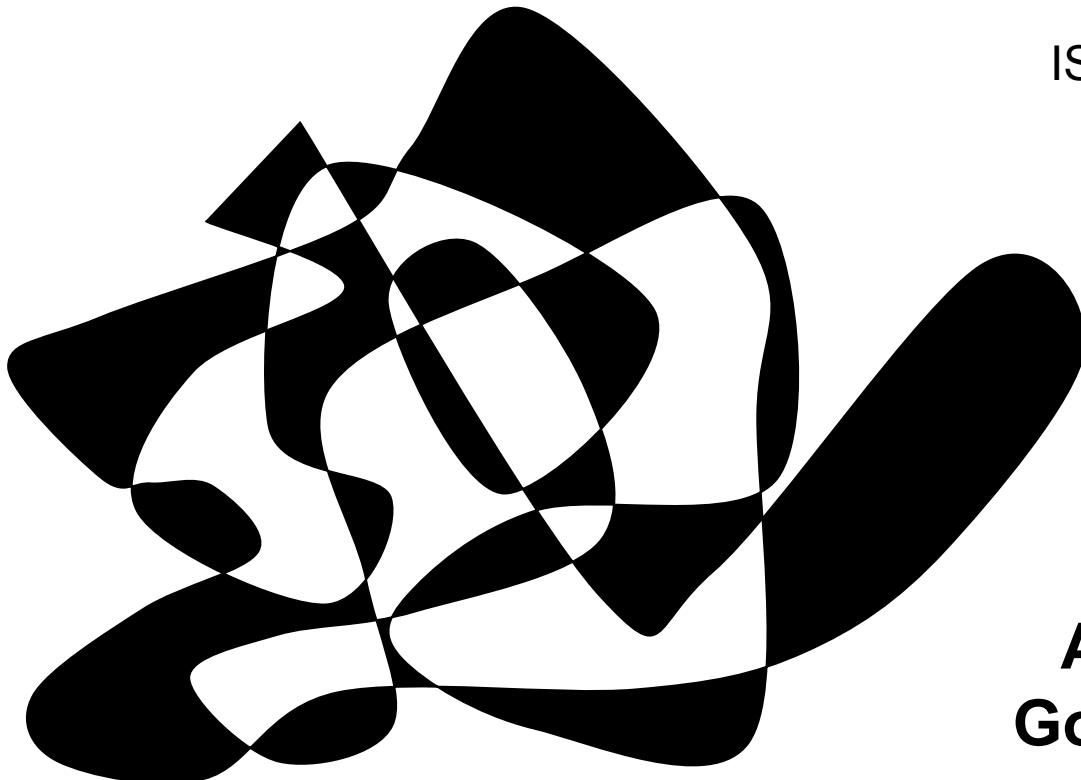


# **Advanced Project Management**

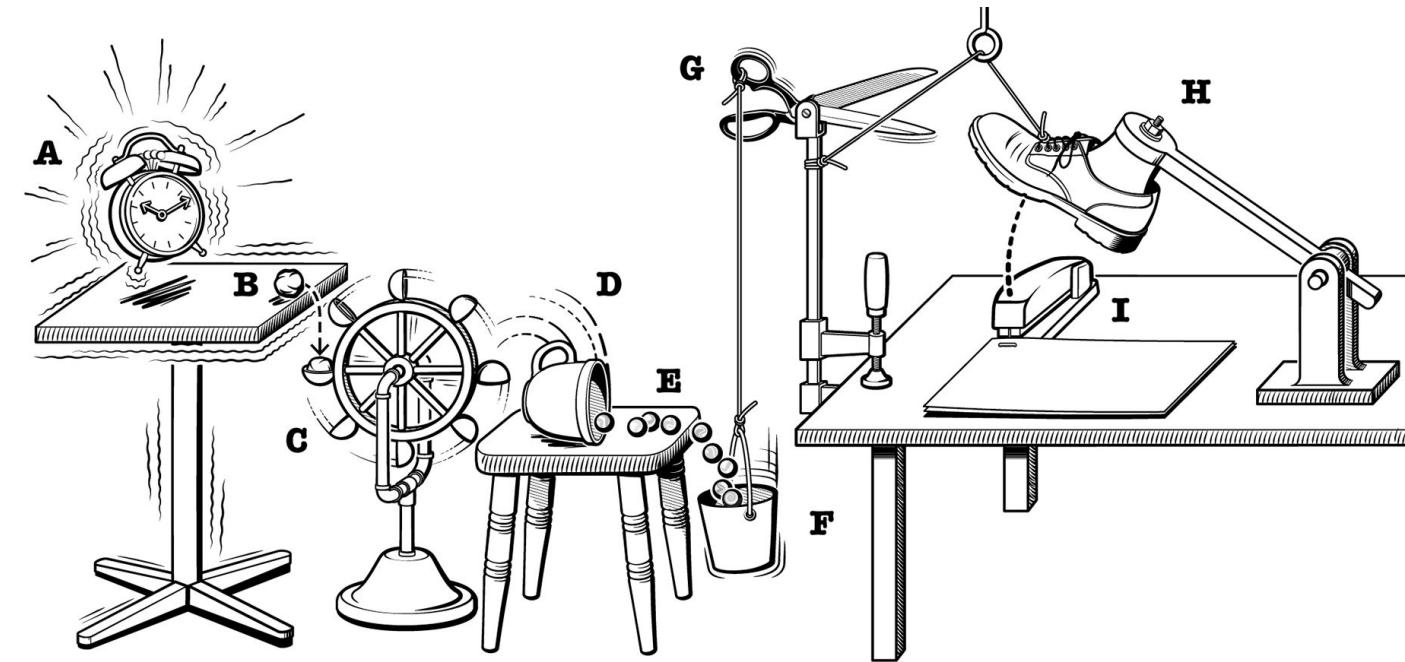
IS 594, Section PJ

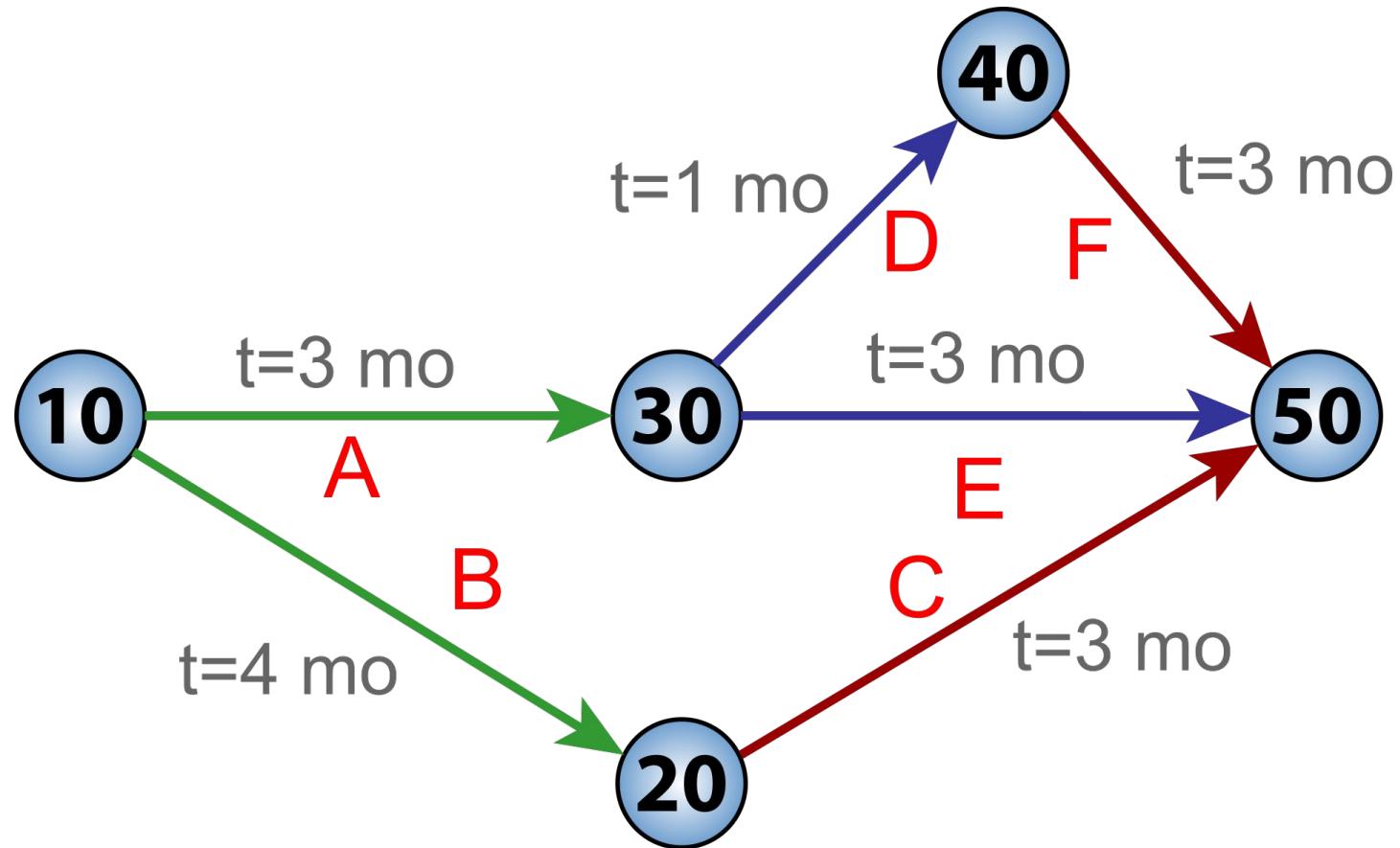


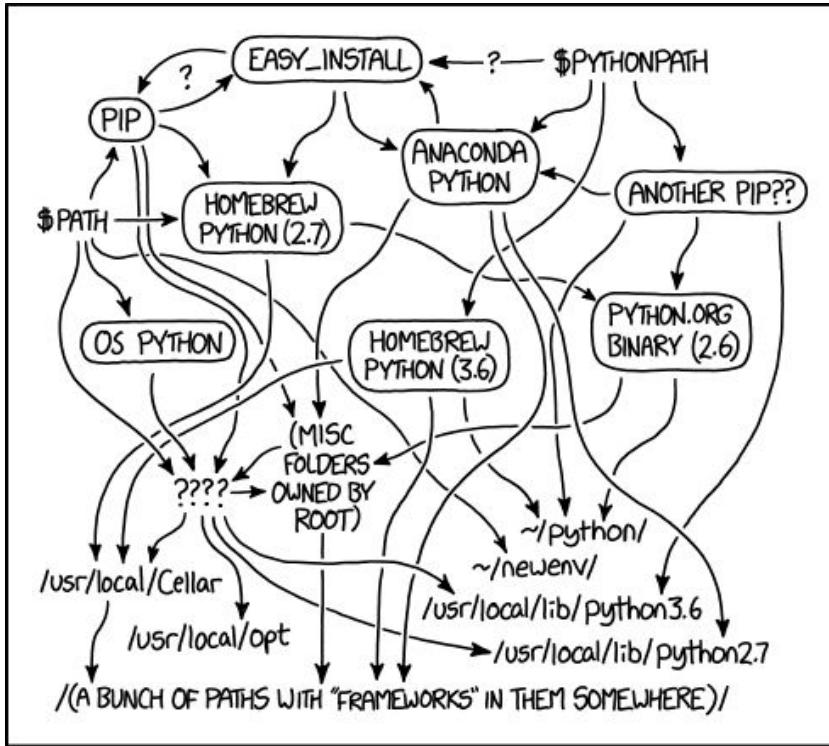
**Applications I (Rube  
Goldberg Management)**

# Process for no Good Reason?

What dependencies are involved? Contingencies provide both opportunities and challenges.







**xkcd to the rescue!**  
**(FTW)**

Python Environment, xkcd: <https://xkcd.com/1987/>

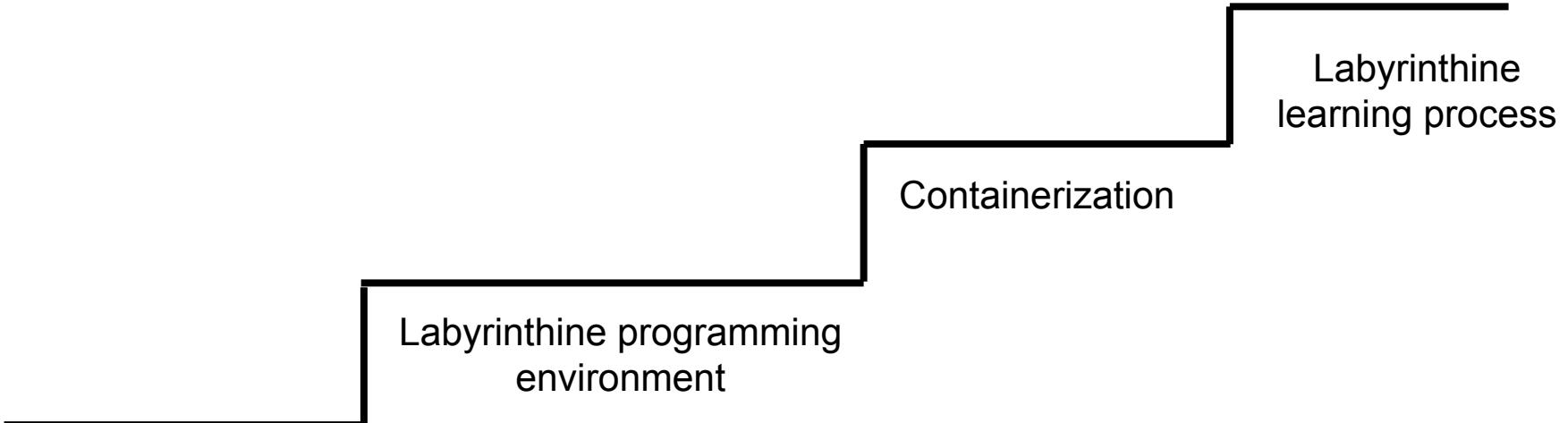
Matteo. [Avoid the Chaos: How to Properly Manage Python Packages with Virtual Environments](#). dev.to blog.

Managing libraries and importing packages in Python. Bureaucracy-inducing, but can be managed. Virtual dev environments can help, but adds a technical barrier.

```
$ tree -L 4 myenv
```

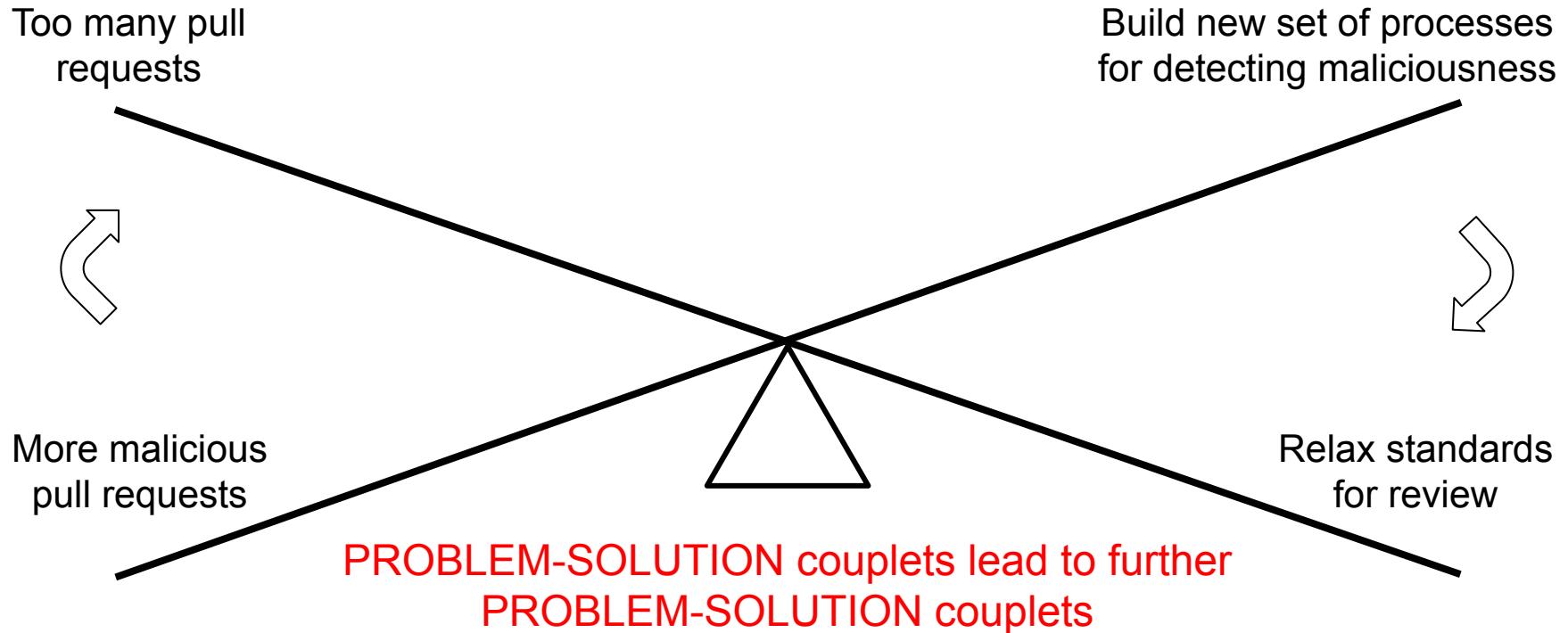
```
myenv
├── bin
│   ├── activate
│   ├── activate.csh
│   ├── activate.fish
│   ├── Activate.ps1
│   ├── pip
│   ├── pip3
│   ├── pip3.10
│   └── python -> python3
        └── python3 -> /usr/bin/python3
            └── python3.10 -> python3
├── include
└── lib
    └── python3.10
        └── site-packages
            ├── _distutils_hack
            ├── distutils-precedence.pth
            ├── pip
            ├── pip-22.0.2.dist-info
            ├── pkg_resources
            ├── setuptools
            └── setuptools-59.6.0.dist-info
└── lib64 -> lib
└── pyvenv.cfg
```

# Ladder of Bureaucracy: where is the Goldilocks rung?



PROBLEM-SOLUTION couplets lead to further  
PROBLEM-SOLUTION couplets

# Ladder of Bureaucracy: where is the Goldilocks rung?



# **Bureaucracy as the proliferation of processes**

Perpetual addition of workflows: every expansion of a process requires a new set of components.

Governance: laws on top of laws: bricolage of rules freezes up organization (Indian License Raj).

Bureaucracy: hierarchies of authority, specialized functions, and formalized processes (all organizations will have bureaucracy).

Stands in contrast to disorganization and improvisation, but is the continual growth of bureaucracy always necessary?

# Can Bureaucracy (organizational control) be Helpful?

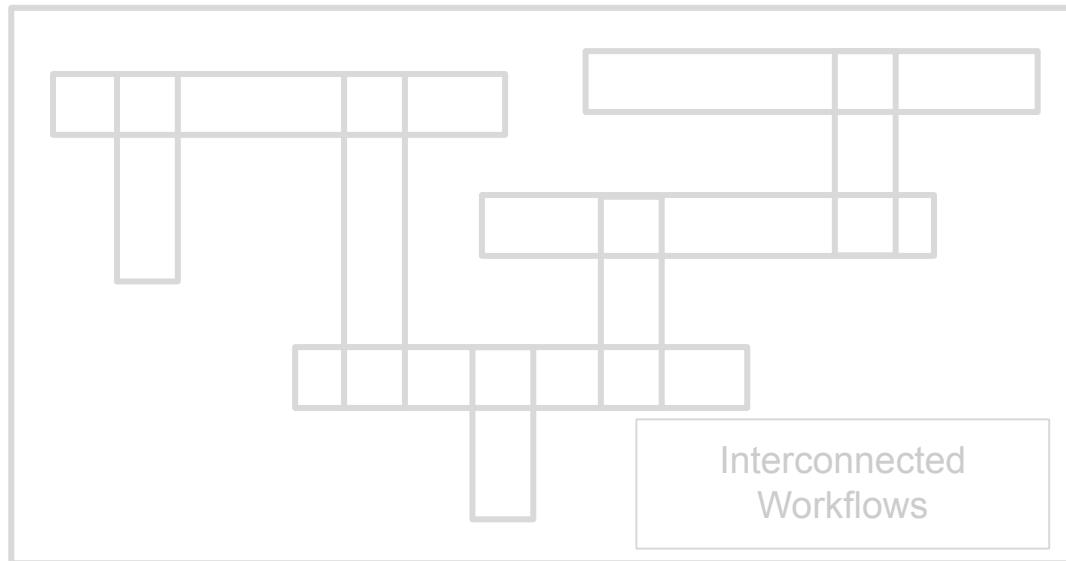
Bechky and Chung (2017). Latitude or Latent Control? How Occupational Embeddedness and Control Shape Emergent Coordination. *Administrative Science Quarterly*, 63(3).

**Equipment manufacturing:** communities visibly conformed to organizational control processes while exercising occupational control behind the scenes to coordinate emergently.

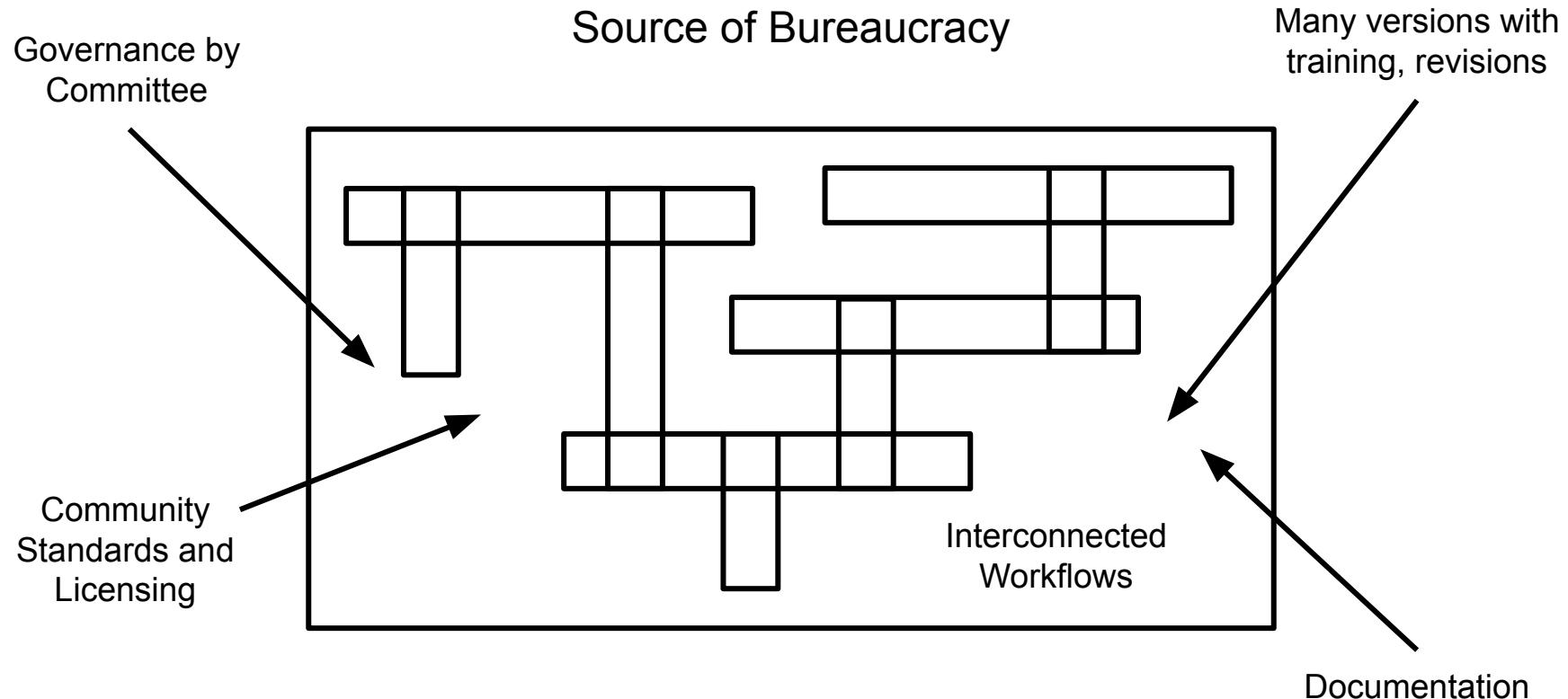
**Film production:** organization granted the occupational community significant latitude over its tasks, which enabled members to coordinate emergently to solve problems the majority of the time.

# Bureaucracy - Technical Debt Analogy

Roadmaps, workflows, community standards, and governance are all forms of bureaucracy. We need a certain amount of bureaucracy, but it should be minimized to a manageable degree.

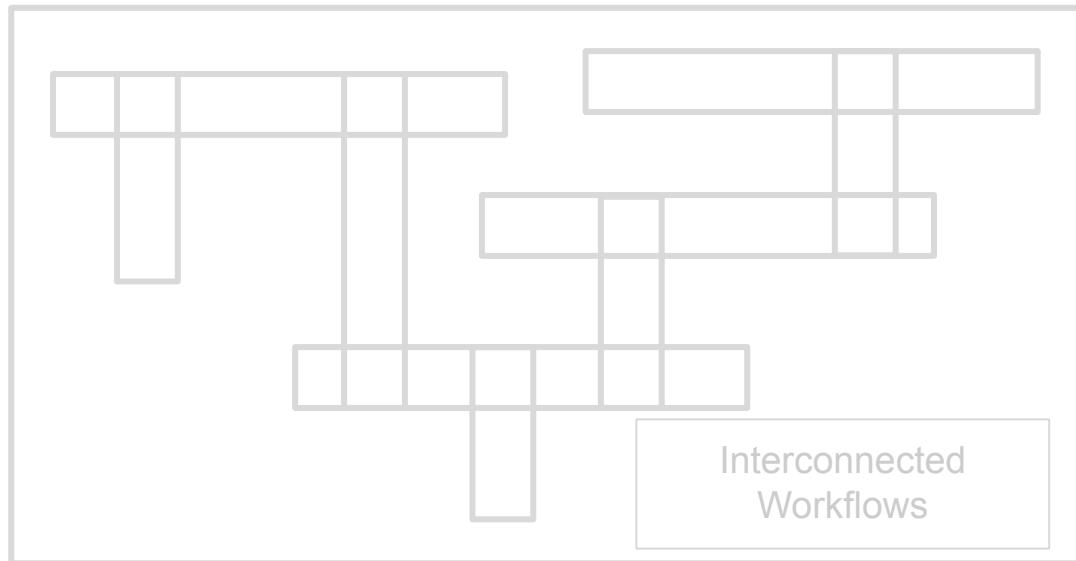


# Bureaucracy - Technical Debt Analogy



# Bureaucracy - Technical Debt Analogy

Short Development Cycles, or massive changes in the market (or world), can make bureaucracy fatal or your organization. “Sclerotic” response time (as bureaucracy impinges an organization’s adaptability).



**Simplicity (and Optimization) is Beautiful?**

# Managing Feature Cycles

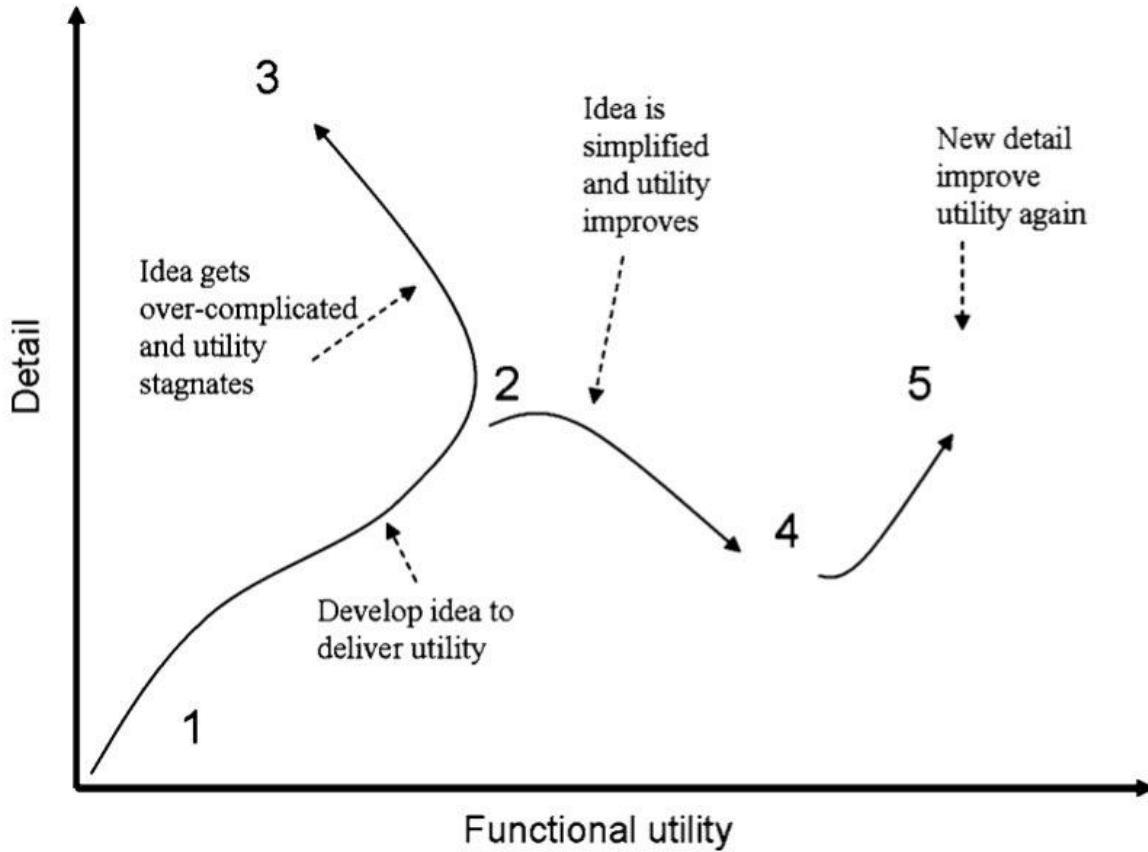
From Chapter 9, “Program Management for Open Source Projects”  
(Ben Cotton).

Manage features as a series of issues and milestones.

- templates: can be used to define scope, testing plan, contingency plan, and rationale.
- scale and approval process: who decides what is included in a formal release? What is too detailed, and what is too trivial.

How are features enforced?

- feature wranglers: open-source leaders or centralized managers.
- feature lifecycle: proposal window → timeline → completion path.



**Fig. 2** The simplicity cycle [adapted from Ward (2005)]

## **Connection between process, rules, issue size, and product features**