# Introducing Python to Government Systems

Robin Friedrich
Python Workshop 5
Washington D.C.
November 5, 1996



## Introducing Python to Government Systems

- Freeware can be a viable option
- Quality of software can be quite high
- Minuses to be aware of
- Approaching Management
- Steps to Technology Insertion
- The example of Python

## New Options for Government Software

- Typically, governments and large organizations have dealt with software as a make/buy decision
- A third option is proving to be viable - Adopt
- In the Unix environment especially, source freeware has been alive and well for some time
  - Very often your problem has been encountered and dealt with by others
  - The openness of these systems (accelerated by the internet) encourages people with good tools to share



## Widespread sharing of source code has several benefits

- Software flaws are uncovered early as it is exercised by many different folks
- Since source code can be examined, flaws hidden from the average user can be noticed and dealt with before trouble happens
- Once the source is understood, it can be tailored to meet local needs
- Portability can be readily tested
- Malicious code is more difficult to hide

## Minuses

- Freeware can be fraught with problems
  - Nobody to help with installation or support
  - Upgrades happen when the author feels like it
  - Development may vanish altogether
  - Works well only on one platform
- Payware is not necessarily better
  - Software quality control suffers from rushed schedules
  - Available only on large market platforms, not your odd platform
  - Companies can be unresponsive to user requests, bugs go unfixed for a long time
  - New releases happen on their schedule, not yours
  - Life cycle cost can be high
  - Companies go out of business too



## You're going to put WHAT on my system?!

- Managers can easily take the stance that all freeware is a bad risk.
- Point out the other side of the coin
  - Payware isn't always a bed of roses
  - Writing your own software is very expensive
  - Cite other successful users
- Argue based on tangibles, avoid an emotional plea
  - Be above-board; don't load the deck by quietly bringing in a neat tool, becoming dependent on it, and then telling your boss you MUST have it forever.
- Have a Process for Technology Insertion!

## Example Technology Insertion Process

- The technology must meet a known need
  - Defends against "hobby shop" accusations
- The technology must be less costly to implement than alternative solutions
  - Life cycle cost / quality analysis must be fairly conducted against alternatives (commercial or otherwise)
  - Make vs. Buy vs. Adopt
- The technology must be maintainable at a low cost either by third party source or with modest in-house effort
  - $$$ talks
- The technology must be available in full source code form



## ...continued.

- The technology must have a thorough test suite to help verify its proper functioning
- The technology must be portable to all envisioned operational environments
  - Don't corner yourself with "*kewl*" software
- The technology must not impose indirect dependencies which may not be available on other target platforms
  - Look at future platform trends and assess how likely it is not to satisfy some critical functionality.
- Check the Copyright
  - Your procurement people must be comfortable with whatever limitations are imposed by the copyright language

## Case in point - Python

- NASA adoption of Python for Shuttle Flight Design.
  - We needed an object oriented scripting language to help integrate a C++ system and provide a high level user interface for engineers
  - Python was a perfect fit but that fact wasn't obvious to management
  - A standardized process was needed to make informed decisions with regard to new technology insertion
  - Python was then racked up against these criteria to prove it's worth



## The Checklist

- ✔ Known need
- ✔ Life cycle cost analysis vs. alternatives
- ✔ Maintenance cost
- ✔ Source code
- ✔ Test suite
- ✔ Portability
- ✔ Dependencies
- ✔ Copyright

## Conclusion

- Freeware available in source form can be safe and reliable for production systems
- Make intelligent decisions about software through an agreed on technology insertion process
- Python is an excellent example of quality software available on the internet that can be put to use with great effect to solve problems many government organizations face.