

how to replace a legacy tool with 100k installations and 50 reverse package dependencies

Serafeim Zankolas <sez@debian.org>

Debian Developer | Head of devops at leemail.me

FOSDEM 2012-02-04



audience

maintainers of community distributions (no BDFL).
case study is Debian-specific, but lessons are generic



take home message

an iteratively updated **design doc** is **key to consensus** for **and testability** of a change with cross-package impact



case study: update-inetd

- perl script and module written in 1994
- is invoked by maintainer scripts of server packages that run via inetd, to manage service entries in inetd.conf
- increasingly irrelevant, like inetd itself, but nevertheless installed in 85% and actually used in 32% of all Debian installations (popcon vote)
- broken by design (uses only service name as key to manage inetd.conf entries)



a tail on a change with cross-package impact

bug report #24043, opened in **1998**:

update-inetd can only add one entry for a service

mail to debian-devel list, **2001**:

RFC: new update-inetd

msg #10 in bug report #179318, **2003**:

this bug will not be investigated because update-inetd is supposed to be rewritten from scratch, some day.



a tail on a change with cross-package impact (cont'd)

msg #91 in bug report #8927 sent in **2007**, opened 1997:

the reason why no conclusion has been reached is because no one has written the code for any of the various reasonable solutions that have been proposed.

mail to debian-devel list, Sept **2009**:

RFC: update-inetd migration to dpkg-triggers

mail to debian-devel list, Feb **2011**:

DEP9: inet-superserver configuration by maintainer scripts



how not to make a proposal with cross-package impact

come up with an idea and send it to the largest mailing list



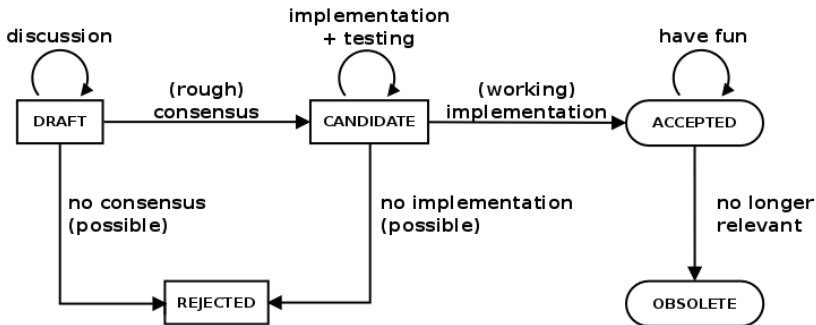
how to make a proposal with cross-package impact

- draft a design doc
- discuss it with someone who won't call you names for having a silly idea
- try a proof of concept implementation
- discuss it with people directly impacted by the changes
- if all goes well, then tell the whole world
- iterate whenever necessary

- people not yelling at you is a good sign
- being ignored until you show some code is normal



a process to keep track of (lack of) progress



see <http://dep.debian.net/> for listing and status of current Debian Enhancement Proposals



design doc excerpt: overview of operation

	server program exists	status of inetd.conf entry	matching reconf-inetd fragment	shadow fragment status	reconf-inetd action
0	no	disabled	no	identical	remove
1	no	enabled	no	identical	remove
2	yes	disabled	yes	different	enable
3	yes	disabled	yes	identical	enable
4	yes	missing	yes	n/a	add
5	commented-out inetd.conf entry				none
6	any other combination				none



design doc excerpt: overview of operation

	context server program exists	context status of inetd.conf entry	context matching reconf-inetd fragment	context shadow fragment status	expectation reconf-inetd action
0	no	disabled	no	identical	remove
1	no	enabled	no	identical	remove
2	yes	disabled	yes	different	enable
3	yes	disabled	yes	identical	enable
4	yes	missing	yes	n/a	add
5	commented-out inetd.conf entry				none
6	any other combination				none



parenthesis: one slide intro to behaviour-driven dev't

- program behaviour is described in terms of **features**
- every feature is described with one or more **scenarios**
- every scenario is a given/when/then sequence of **steps**:

Given context
When event
Then expected outcome

1: Describe behaviour in plain text

```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

Scenario: Add two numbers
  Given I have entered 50 into the calculator
  And I have entered 70 into the calculator
  When I press add
  Then the result should be 120 on the screen
```

2: Write a step definition in Ruby

```
Given /I have entered (.*) into the calculator/ do |n|
  calculator = Calculator.new
  calculator.push(n.to_i)
end
```

3: Run and watch it fail

```
$ cucumber features/addition.feature
Feature: Addition # Features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers # Features/addit
Scenario: Add two numbers # Features/addit
  Given I have entered 50 into the calculator # Features/step_d
  uninitialized constant Calculator (NameError)
  /features/step_definitions/calculator_steps.rb:2:in `given /
  Features/addition.feature:7:16 - Given I have entered 50 into
  And I have entered 70 into the calculator # Features/step_d
  When I press add # Features/addit
  Then the result should be 120 on the screen # Features/addit
```

4. Write code to make the step pass

```
class Calculator
  def push(n)
    @args ||= []
    @args << n
  end
end
```

5. Run again and see the step pass

```
$ cucumber features/addition.feature
Feature: Addition # Features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers # Features/addit
Scenario: Add two numbers # Features/addit
  Given I have entered 50 into the calculator # Features/step_d
  And I have entered 70 into the calculator # Features/step_d
  When I press add # Features/addit
  Then the result should be 120 on the screen # Features/addit
```

6. Repeat 2-5 until green like a cucumber

```
$ cucumber features/addition.feature
Feature: Addition # Features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers # Features/addit
Scenario: Add two numbers # Features/addit
  Given I have entered 50 into the calculator # Features/step_d
  And I have entered 70 into the calculator # Features/step_d
  When I press add # Features/addit
  Then the result should be 120 on the screen # Features/addit
```



reuse the design doc for system tests

Feature: service removal

Scenario: service removal-01

Given an inetd.conf file with an enabled entry ftpd_ssl,tcp,/usr/sbin/ftpd_ssl

And a matching server file that does not exist

And no matching reconf-inetd fragment for the service ftpd_ssl,tcp,/usr/sbin/ftpd_ssl

And a matching shadow fragment with identical server arguments for the ftpd_ssl,tcp,/usr/sbin/ftpd_ssl

inetd.conf entry

When I run reconf-inetd

Then the ftpd_ssl,tcp,/usr/sbin/ftpd_ssl service entry is removed from inetd.conf

And the matching shadow fragment with identical server arguments for ftpd_ssl,tcp,/usr/sbin/ftpd_ssl is removed

And inetd is restarted

Scenario: service removal-02

Given an inetd.conf file with a maintainer_disabled entry ftpd_ssl,tcp,/usr/sbin/ftpd_ssl

And a matching server file that does not exist

And no matching reconf-inetd fragment for the service ftpd_ssl,tcp,/usr/sbin/ftpd_ssl

And a matching shadow fragment with identical server arguments for the ftpd_ssl,tcp,/usr/sbin/ftpd_ssl

inetd.conf entry

When I run reconf-inetd

Then the ftpd_ssl,tcp,/usr/sbin/ftpd_ssl service entry is removed from inetd.conf

And the matching shadow fragment with identical server arguments for ftpd_ssl,tcp,/usr/sbin/ftpd_ssl is removed

And inetd is restarted



write a program to write all scenaria for you

```
1 for conf_entry_status in conf_entry_status_values:
2     for server_path in server_path_values:
3         for reconf_fragment in reconf_fragment_values:
4             for shadow_fragment_status in \
5                 shadow_fragment_status_values:
6
7                 # service addition scenarios
8                 [..]
9                 # service removal scenarios
10                [..]
11                # service enable scenarios
12                [..]
13                # no action scenarios
14                [..]
```

translate scenario steps to code

```
1 @step('a matching server file that exists')
2 def a_matching_server_file_that_exists(step):
3     server_path = world.scc.service.get_server()
4     server_dir = os.path.dirname(server_path)
5     os.path.exists(server_dir) or os.makedirs(server_dir)
6     open(server_path, 'w').close()
7
8 @step('a matching server file that does not exist')
9 def assert_missing_server_file(step):
10    server_path = world.scc.service.get_server()
11    assert not os.path.exists(server_path)
```



make the tests pass

```
feature: service addition
Scenario: service addition-01
    Given an inetd.conf file with a missing entry ftpd_ssl,tcp,/usr/sbin/ftpd_
    And a matching server file that exists
    And a matching reconf-inetd fragment for the service ftpd_ssl,tcp,/usr/sbi
    When I run reconf-inetd
    Then a new entry is added to inetd.conf for service ftpd_ssl,tcp,/usr/sbin
    And a matching shadow fragment with identical server arguments for ftpd_ss
    And inetd is restarted
4 features (4 passed)
40 scenarios (40 passed)
244 steps (244 passed)
```



behavioural driven development test tools

<http://lettuce.it> (python)

<http://cukes.info> (ruby)

DEP9

<http://dep.debian.net/deps/dep9> (design doc)

<http://git.debian.org/?p=collab-maint/reconf-inetd.git> (code)



some words of wisdom from the DPL

<liw> what's your foremost suggestion for making Debian development more fun for everyone?

<zack> be ***bold***!

<zack> **spot something that needs improvement**

<zack> **take responsibility** for improving that

<zack> and **do** that

