AT&T Software Symposium 2011
Proposal (Long Talk)

# *Ruby tools for writing (and deploying) software that matters*

Mani Tadayon
Senior Software Engineer
AT&T Interactive
mtadayon@attinteractive.com

One of the strengths of the ruby development community is the proliferation of open-source tools to encourage and facilitate best-practices throughout the software development process. I will demonstrate an example of using *Cucumber, RSpec, RVM, Bundler, Puppet, Vagrant* and other ruby tools to develop a non-trivial application at lightning speed, while ensuring excellent documentation, integration testing, acceptance testing, unit testing, modularity, verifiability, production-readiness, and most importantly, programmer joy.

I will also attempt to propound a holistic development philosophy, that goes beyond a collection of tools. The core of this approach to development is rooted in *Behaviour-Driven Development*, which evolved out of *Test-Driven Development*, and which strives to write software that matters (and nothing but). One of the challenges to getting the maximum benefit from the numerous ruby tools available is to understand how they can contribute to a larger framework for development. Although each library or tool has its own domain of usage and its own peculiarities, I will attempt to show that a remarkable consistency exists between them. Further, once this consistent dedication to simplicity, verifiability and automation is understood (and embraced), it will become natural to use multiple sophisticated tools in concert.

*Cucumber*, the flag-bearer for *Behaviour-Driven Development*, is a tool for creating high-level specifications in plain English using a rigid grammar. A typical Cucumber specification could read:

```
Feature: Pay bills online
      As an AT&T customer
      In order to save time and trees
      I want to be able to pay my bill online

Scenario: Make payment with existing payment method
Given I am on the “Manage My Account” page
And I have a valid payment method on file
When I press “Pay My Bill”
Then a payment should be made for the current bill
```

Cucumber simultaneously addresses documentation, acceptance testing, regression testing and integration testing, while allowing non-technical stakeholders to engage in the design and specification with high-precision.

Once high-level specifications are ready for a *Minimal Marketable Feature*, we can quickly move to build, test and deploy a fully functional application *with only this single feature*. *RVM* and *Bundler* will be used to specify environmental and library dependencies, respectively, for the application. These dependencies can then be systematically specified using *Puppet* so that we can launch and configure a machine, physical or virtual, at will to deploy and test the application. *Vagrant* will be used to integrate configuration of a deployment architecture with the development process itself. The use of *Puppet* & *Vagrant* front-load the pain of deployment architecture and will also detect any gotchas early-on, so that we can re-architect the application as it is being developed, not at the last second before deployment to production.

Finally, once a working deployment architecture is ready, we can begin to write our code using *RSpec* and *Cucumber* in tandem.