The road from good software engineering
to good science
...is a two way street...
Three Themes:

Philosophy
Interlude on Goodness
Lessons from Science
Philosophy
Good
Good as in Quality
Fundamental Premise
Our community needs to think more about science, and about being able to reproduce results, and formulate theories that let us make predictions about language.
The key to making that happen is making our software and data more usable, more available, and making such acts of sharing more central to our field.
If we do that, our software engineering is pretty good
Science
Develop theories or models that let us make predictions about the world
Our world is language...
Good Science
... are those methods that result in experimental findings that an independent observer can reproduce
Good Software Engineering
...are those methods that result in software that anyone can use, anytime, anywhere...
...to reproduce our results...
Experimental results that you publish are the test cases for your ideas.
...and your software...
Can't discount the role of software
...although many try...
“It's really the ideas that count...”
“Well, the algorithm is described in the paper...”
“It's really just a prototype...”
“Well, I got a new computer and I don't think the software made it to the new one...”
“Ummm ... my student left and I don't quite know how he did all this...”
Unacceptable
I did this experiment on X
Here are the results...
Accept them
No, the software isn't available
Neither is the data
I simply assume you have 8 months available to reinvent my method
And that you can do that from an incomplete description
Cheers!
That's many things ...
It's not science
Empiricism is Not a Matter of Faith
Computational Linguistics
September 2008
Software and NLP
Good Software
Should Work
Anytime
Anywhere
For Anyone
...and it should certainly work for you 6 months in the future
...or 5 years from now...
... it should work for others today, and 5 years from now ...
...even if you've moved on, aren't answering email, and the project is over
If your software can do that, it's pretty well engineered
Will your software work in 40 years?
You should hope so ...
Make choices that make that at least possible
Think of your software as a time capsule
Think of it as your chance for immortality
How many hours have you spent away from loved ones, friends, adventure, nature, romance, and life ...
... to create, test, and use software?
At least make it last...
Let someone 100 years from now unpack your code and data, and be able to read it, understand it, run it, and modify it.
Let yourself be able to do the same thing in 10 years
If your software can do that, it's pretty well engineered
Will the Linux Kernel be available and running in X years?
There's a good chance
Company won't go out of business
ANSI C will be around for a long time
Virtualization will keep architectures alive even when hardware is gone
Make choices that give your code (and your legacy) a chance too
Don't rely on the newest priceiest weirdest goofball proprietary bleeding edge hardware and software
The NeXTcube is a versatile, easy-to-use workstation that can be utilized as a desktop monochrome system, true color 32-bit-per-pixel color/video workstation or file server system, all featuring NeXT’s object-oriented operating and development environment.

Whether used with the NeXTdimension™ board as a standalone workstation incorporating 32-bit-per-pixel color/video, or as server on a network, the NeXTcube™ computer offers a tremendous amount of flexibility and performance in a single, one-foot-square magnesium cube. The system is built around the Motorola 25-megahertz 68040 CPU with integrated memory management and floating-point units, and includes the Motorola 56001 Digital Signal Processor for superior sound handling.

The NeXTcube may be equipped with 16 to 64 megabytes of main memory, and offers a variety of storage options—ranging from a 2.88-megabyte floppy disk drive to hard drives with capacities from 400 megabytes to 2.8 gigabytes. In addition, there are three available NeXTbus™ slots, so additional functionality can be added to the NeXTcube via NeXTbus expansion cards from third-party vendors or from NeXT, making the NeXTcube an extremely versatile workstation.
Don't hoard
Take advantage of public repositories which likely endure and proliferate
Think about who is included in your definition of “anyone”
...with $200?
...with $20,000
...with a PhD in Computer Science?
...and a staff of 10?
...with 4 weeks available to debug?
...and another 6 months to reimplement?
Interlude on Goodness
No matter how well engineered our software is ...
Life will be hard and a bit cruel for many ...
So be a little humble
Appreciate your good fortune
And push yourself a little harder
Think about what you can give back to the scientific community
Think about the people who fund your work
... and I don't mean government project managers, legislators, or corporate titans
Appreciate our good fortune
Live up to the trust that is given us almost without question
And make sure we end up making some progress
Good Science
Produce theories that make reliable predictions about the world
Experiments are described in such a way that the results can be conveniently and reliably reproduced.
Anytime
Anywhere
By Anyone
Gravity
A Good Theory
Works now now
Will work in 10 years
Works here
Works on the moon
Works for me
Works for you
Gravity is a force, not an artifact
Telescope
Works anytime, anywhere, for anyone
The old ones still work
We share the big ones...
If we have access to the same resources, we can reproduce each other's results.
We need to work a lot harder (and engineer systems a lot better) to make that happen.
Not convinced?
Conduct the following experiment
Randomly select 1 of your papers
Reproduce your results
If you can't...
Do you think anyone else can?
What if nobody could have reproduced Galileo's falling objects experimental results? Would we simply believe?
They barely believed him at the time
If your software can reproduce your results, its pretty well engineered
Lessons from Science
We don't get it right the first time
If I have seen further it is only by standing on the shoulders of giants
(who were mostly wrong)
"Reports that say that something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns -- the ones we don't know we don't know."
We don't get it right the first time
Aristotle
(384 – 322 BC)
There are 4 elements
The heavens are different
Different rules apply
Before the telescope, the heavens really were different
Other planets were balls of fire, like the stars, like the sun
Ptolemy
(90 – 168)
Crazy?
Very reliably predicts the movement of heavenly bodies
Instrumentalist
A theory that reliably explains and predicts the existing data
Realistic
A theory that describes things as they “really” are
لا أعلمني لأولى لليأس إلها، مأوى وشلاله كم، وعزمي

السياحة الصغيرة

المير

المير

المسند

المير

اللخواص

المير

اللخواص

المير

المير
Copernicus
(1473 - 1543)
Nicolaus Copernicus
Celebre Astronome, Mathématicien, Philosophe et Médecin, né à Thorn, Ville de la Prusse Royale, mort en 1543, âgé de 70 ans

Copernicus se levant au-dessus du Vulgaire,
Présente à l'univers une nouvelle Sphère,
Et par un effort sans pareil
Rend la Terre mobile et fixe le Soleil.
Wasn't much of an observer
Found Ptolmey's model overly complicated
Wanted a simpler explanation
...that was more heavenly
Came up with another model that was consistent with Ptolmey's data
Great!
(Well, better)
Uniform Motion
Perfect circles
NICOLAI COPERNICI
CENENSI DE REVOLUTIONIBUS ORBrium COelestium
Libri VI.

IN QUAEVS STELLARUM ET Ph.
EADUM ET TERRARUM MOTU, EX VITIS
HIBERUS EYESHOBAS ABHABIBOB, RIGOMI
HABERVS SBETAS EPHEBORAM HABITIBAM, EX QUIBVS
HIBERVS STELLABVS ET TERRABVS, HABEBAM
HIBERVS MATERDVS, EPHEREBAM, HABEBAM
HIBERVS STELLABVS ET TERRABVS, HABEBAM
HIBERVS MATERDVS, EPHEREBAM, HABEBAM
HIBERVS STELLABVS ET TERRABVS, HABEBAM
HIBERVS MATERDVS, EPHEREBAM, HABEBAM
HIBERVS STELLABVS ET TERRABVS, HABEBAM
HIBERVS MATERDVS, EPHEREBAM, HABEBAM
HIBERVS STELLABVS ET TERRABVS, HABEBAM
HIBERVS MATERDVS, EPHEREBAM, HABEBAM
HIBERVS STELLABVS ET TERRABVS, HABEBAM
HIBERVS MATERDVS, EPHEREBAM, HABEBAM
HIBERVS STELLABVS ET TERRABVS, HABEBAM
HIBERVS MATERDVS, EPHEREBAM, HABEBAM
HIBERVS STELLABVS ET TERRABVS, HABEBAM
HIBERVS MATERDVS, EPHEREBAM, HABEBAM

IN BASILIAE, EX OFFICINA
HENRICI PETRINI.
Tycho Brahe
(1546 - 1601)
A great observational astronomer, the last naked eye astronomer
Galileo
(1564 - 1642)
1609 Telescope
1610

Observed 4 moons of Jupiter
Back to Tycho
Made remarkably accurate observations for 20 years
Knew about Copernicus
Arrived at his own theory
The Tychonic Model
A hybrid model
Fits and predicts the observed data
Data Sharing
Kepler
(1571 - 1630)
Why are there 6 planets?
Why are they so positioned?
Geometry and Perfect Solids
Tabula III Orbium Planetarum dimensiones, et distantias per quinque regularia corpora geometrica exhibens.


Tensregio, et Tetcio, Comiti Montis Belgarum, etc. consecrata.

295. Kepler. 1596. (Greatly reduced.)
In 1601 Tycho bequeathed his data...
Kepler's Laws of Planetary Motion
Varying velocity
Elliptical Orbits
...around the Sun
It was left to Newton to work out what held the planets in place and made them move...
History of Science?
We are wrong many many times before we are right
Progress happens when people leave their data and instruments behind.
Ptolemy (90 - 168)
Copernicus (1473 - 1543)
Tycho (1546 – 1601)
Galileo (1564 - 1642)
Kepler (1571 - 1630)
Newton (1642 - 1727)
Good science and good software assume you don't get it right at first
Leave your software (and your data) behind for your successors to build on
And if they can, you've done some good software engineering, and some good science
Ted Pedersen

University of Minnesota, Duluth
http://www.d.umn.edu/~tpederse
tpederse@d.umn.edu