Lessons and Stories from My Career

Gary T. Leavens
ECOOP Doctoral Symp.
July 5, 2006
Caveat

- Your mileage may vary
- Not me:
Goal: Positive Impact

What work of mine do you know?
Story

Being introduced at OOPSLA
Teaching Tip

- If you want interaction:
  - Get it on day 1
  - Wait for it...
Fundamental Problems for Making a Positive Impact

- Lots of prior work
- Lots of researchers
Ways to Make a Positive Impact

- Publish important work first
  - Think hard
  - Use new techniques/instruments
  - Work in underdeveloped area
  - Start new (sub-)area
- Publish clear descriptions
  - Relate to current understanding
- Be persistent
How to Improve?

- Ask a lot of questions
  - “Why?”
- Read a lot
- Develop judgment about
  - Problems
  - Solution techniques
  - Explanations
As Undergraduate

Read books about computing
- Computer Lib!
- Gödel, Escher, Bach: An Eternal Golden Braid
- Art of Computer Programming, ...
- Programming Language Reference Manuals
- CACM

Now:
- Other (Computer, CACM, TOPLAS, ...)
- Scientific American, Sky and Telescope
- Science fiction
- Science
Lessons?
Lessons?

Reading helps with:

- English (writing), science style
- Seeing problems
- Ways of thinking
- Developing context (general understanding)
- How to explain to outsiders
- Telling a story

Reading fast helps
Reading Math

- Reading ahead in Jr. High algebra
- At MIT:
  - Math minor
  - Commutative Algebra
    - Finding examples
    - Why?
    - Finding readings for context
What does that say about Learning?
Lessons for Learning?

- Motivation important
- Context helps learning
- Concrete examples help
- Can’t just have examples

- Think about how you learn things...
Writing my Dissertation

- Guttag’s advice: “Keep a list”
- Hardest lesson: “Don’t core dump”
- Writing is like programming
# How is Writing Like Programming? (Theory Version)

<table>
<thead>
<tr>
<th>Written Text</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>Declaration</td>
</tr>
<tr>
<td>Theorem statement</td>
<td>Procedure interface (specification)</td>
</tr>
<tr>
<td>Proof</td>
<td>Implementation</td>
</tr>
<tr>
<td>Lemma</td>
<td>Subroutine</td>
</tr>
<tr>
<td>Remark</td>
<td>Comment</td>
</tr>
<tr>
<td>Example</td>
<td>Test case</td>
</tr>
</tbody>
</table>
## How is Writing Like Programming? (Systems Version)

<table>
<thead>
<tr>
<th>Written Text</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>Declaration</td>
</tr>
<tr>
<td>Goal (or problem)</td>
<td>Procedure specification</td>
</tr>
<tr>
<td>Description of Code</td>
<td>Implementation</td>
</tr>
<tr>
<td>Subproblem</td>
<td>Subroutine</td>
</tr>
<tr>
<td>Application/Example</td>
<td>Comment</td>
</tr>
<tr>
<td>Performance results</td>
<td>Test case</td>
</tr>
</tbody>
</table>
Bill Weihl’s Recommendation

  - See *Style* by Williams *(U. Chicago, 1990)*
Non-linking

Sentences have 2 parts in English.

Links to previous material appear in the first part.

Emphasis and new information are provided by the second part.
Linking idea

In English sentences have 2 parts.

The first part links to previous material.

The second part provides new information and emphasis.
Other Writing Ideas

- Illustrate with examples
  - Also counterexamples!
  - Especially anything initially unclear
- Honesty
  - Present facts, don’t sell
  - Look for flaws
- Later:
  - Writer’s workshops
  - “Pair writing” with students
Writing Related Work

- Related to problem
  - Not just to your solution technique
- Help reader fit your work into problem space

- Say how helps solve problem
- Say why / how doesn’t solve problem
  - Also how solution techniques differ
Getting All the Related Work

- Read other dissertations
- Ask the experts
- Read the references in good papers
  - Science Citation Index
  - Recent conferences / journal issues
- You may need to go to the library!
- Peters and jmlunit story
Good Writing is Revising

  - Revised TR (1989)
  - And main article from it (1993, published 1995)
- ECOOP 2005 article story
How I Practice Writing

- Homework, tests
- Technical reports
- Referee reports
- E-mail!
  - Write it
  - Revise it!
Why Spend Time on Email?
Why Spend Time on Email?

- Helps with peer contacts
- It’s practice
- Doing things well is rewarding (Kierkegaard)
Finding Good Ideas

- Look for problems
  - In reading, teaching
  - By using your own tools / systems
- Have lots of ideas
- Pursue ones that:
  - You are uniquely qualified to handle
  - Tackle important problem
  - Excite you
  - You make progress on
Finding My Thesis Problem

- OO programming new (Liskov)
- Talked to expert programmers
- Informal descriptions unsatisfying
- How to formalize
  - Theorems compare 2 things
  - Wanted airtight explanation
  - Lessons need experiments
Lessons?
Lessons?

- **Focus on problems**
- **Look for what is surprising / new**
- **Think about the end result**
  - Theorem
  - Experiment
Peer to Peer Networking

- Peers contain next generation of top computer scientists
- Get to know them
  - Conferences, workshops, etc.
  - Read papers
- Share ideas
Collaboration Stories

- Jeannette Wing
- William Cook
- Craig Chambers
- Todd Millstein
- Don Pigozzi
- Rustan Leino
- Peter Müller
- David Naumann
Lessons?
Lessons?

- Scientists don’t (usually) bite!
- Collaboration helps one’s career
- Easiest: collaborate with your cohort
- Mentoring is a contribution
- Helps keep you on cutting edge
Egoless

- At MIT: always someone better
- Do your best
  - With resources / time you have
  - Best ≠ perfect
- Strive for improvement
  - Mistakes – correct them
  - What new skills would help?
Liskov and Wing 94

- Liskov’s OOPSLA talk
- Liskov and Wing’s 1993-94 papers
What Mistakes?

- Grinding out technical details
- Not getting ideas / concepts published quickly enough
- Missing collaboration opportunity
Resource Management

- Work steadily
- Plan for deadlines
- Have reserve for deadlines
- Learn to say “no”
- Pick few service duties
  - Do good job in them
  - You have excuse while young
If you want something done...

- Introductory course (Scheme) story
Lessons?

- You’ll have to do it
- Pick your battles
  - Important, interesting
  - Something you can learn from
- Be sure it makes an impact
- Doing a good job takes time
  - So don’t do too many of them
- Eventually you may have to move on
My Refereeing and Reviewing

![Chart showing refereeing and reviewing over time]

- Jour.
- Con. PC
- WS PC
- Prop.
Lessons

- Limit to how much can / want
- Do an excellent job
- Helps make your reputation
- Contributes to community
Tips for PC Members

- Get expert help for top conferences
  - Ask quickly
  - Read the paper yourself also
  - Only for papers that may be good
- Don’t spend too much time on bad
- Look for the good
- Hard part: getting papers in
  - Make the case: contributions, evaluation
- Comments on related work need citation!
Summary

- Goal: make a positive difference
- Read widely
- Strive to improve writing
  - Writing is like programming
  - Write to explain and understand
- Generate ideas and pick best
- Network with your peers
- Do your best