How to Survive in the “Real World”

Tales of a Physicist Turned Software Engineer

Michael Ogg

Founder and Chief Technology Officer, Valaran*

ogg@galibier.net

April 14, 2006

*RIP May 2000-February 2006

Part I Skills Needed

Part II Life in a Big Company

Part III Life in a Small Company

Part IV So you want to start a company?

Part I Skills Needed

- Software Engineering:
  - The process of producing software
  - Much more than programming
  - Much more than computer science
- Roughly, managing the software life cycle

- What is the product?
- Requirements – what should it do?
- Design - architecture, technologies
- Relation to other internal products
- Use of 3rd party products
- Time estimation
- Programming

(cont.)

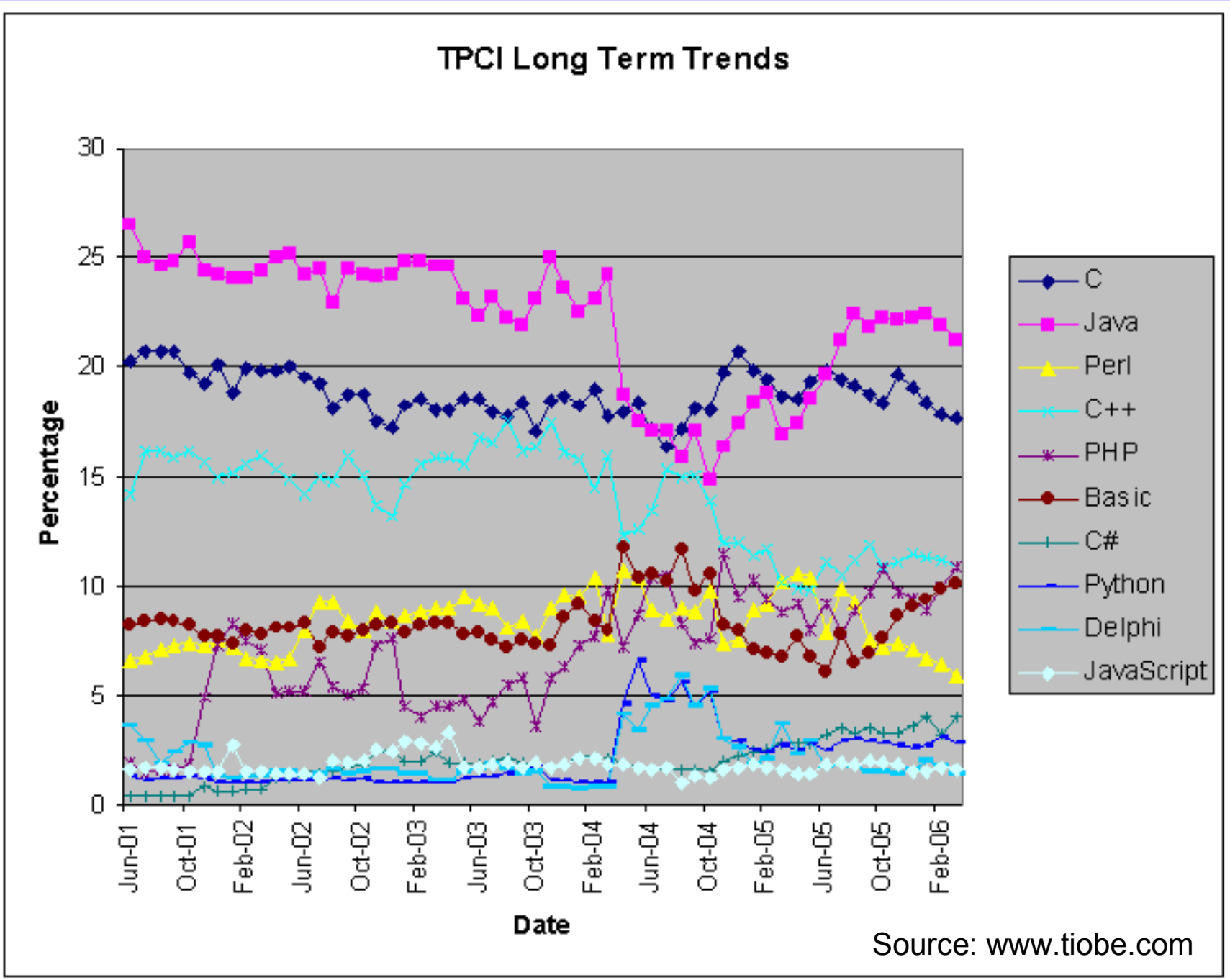
- Testing - unit, system
- Mid-course corrections - ship date, features
- Documentation
- Product release
- Maintenance
- Version management – minor, major
- Product retirement

plus a few feedback loops

- Also need management skills:
  - Managing people (as supervisor or team lead)
  - Managing management (expectations, schedules)
  - Only one thing worse than bad news: surprises
  - Bad news manageable – as long as it's not surprise

Software Skills

- Languages: Object Oriented (mostly)
  - Java almost a “must have”
  - C still (surprisingly) popular
  - C++ makes top 3
- Other:
  - C# growing only slowly
  - Basic (VB) holding around 10%
- Scripting:
  - PHP now the biggest


TPCI Long Term Trends
Percentage
30
25
20
15
10
5
0
Jun-01
Oct-01
Feb-02
Jun-02
Oct-02
Feb-03
Jun-03
Oct-03
Feb-04
Jun-04
Oct-04
Feb-05
Jun-05
Oct-05
Feb-06
Date
C
Java
Perl
C++
PHP
Basic
C#
Python
Delphi
JavaScript
Source: www.tiobe.com

Open source is hot:

- Linux
- MySQL
- GNU C, C++
- CVS
- PHP
- Sourceforge, etc.

partly because of cost, partly overcoming prejudices

- Methodologies
  - Come and go
  - UML currently hottest
- Distributed Computing
  - CORBA all but dead
  - RMI still around, Jini never took off
  - Web Services ruling (sigh)

    see: http://www.jini.org/webinar/1204/Webinar1204.html

Trends:

- Javascript, HTML

  Tho you really don't want to do web development
- AJAX (Asynchronous JavaScript And XML)
- Meta-programming: (e.g. Google Earth)
- Mobile:
  - Java, BREW
- Web Services:
  - Horrible paradigm but toolkits will hide most of it


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Map Satellite Hybrid
Martians landed here
Cranbury Rd
Clarksville Rd
N Mill Rd
Steele Dr
Dean Ct
Grove Mill Pond
Google
Map data ©2005 Tele Atlas - Imagery ©2005 State of New Jersey - Terms of Use

Part II Life in a Big Company

Not that different from a faculty position:

- Generate own ideas
- Get funding
- Build team
- Market inside company
- Manage production

- Big companies are sometimes dysfunctional
  - Cannot make decisions
  - Or cannot execute on decisions
  - Or internal rivalries between divisions
  - Or do not understand core competencies
- And sometimes cannot stick with things
- Beware the reorg!

- Hubris:
  - Because we are Acme Inc. we can do this
  - A company's strength is combination of individuals' strength and company culture
  - Culture is probably more important
  - Seen companies with very talented individuals ground down by poisonous culture

Part III Life in a Small Company

- Build team
- Manage production
- Be able to do part of all other jobs:
  - With CEO raise funds
  - With CMO create marketing message
  - With Sales lead tech discussions
  - With VP Engineering motivate team
- But you could come in in the morning and be gone by the afternoon

- Need to be flexible
- Some customers can be – well, difficult
- You'll wear many hats
- Sometimes tension between sales/marketing in describing company's capabilities

Schedules and Dates

- Dates are *extremely* important - Not that in the big picture they're necessarily so important, but:
  - A missed date gives a customer license to kill
  - Missed dates erode (non-technical) management's confidence in engineering
  - Missed dates erode investors' confidence in company
- Good time estimates are difficult – but essential
  - Sales want aggressive times
  - Engineers want conservative times

Quality

- QA should be decoupled from engineering
- QA's job only to report failure in testing
  - Testing should exercise all aspects of requirements
  - Difficult to get marketing/engineering balance
- Ideally, testing completely automated
  - But rarely is
- “In principle,” any change requires retesting
  - Unit testing relatively straightforward
  - System testing usually more involved

Performance

- Marketing rarely understands quantitatively
  - (Good at buttons here, colors there)
- But for customers, often first consideration:
  - What throughput?
  - How does it scale?
  - What are limits?
- And often hard to prove
- Can be used as “non engagement” rationale

Product vs. Project

- Ideally, produce product and sell it many times
- But product rarely exactly what customer wants
- Customization can be expensive:
  - Uses engineering resources
  - Working on project holds up product
  - Rarely reusable (specialized, IP)

How Often to Release New Versions?

- Why new versions?
  - Bug fixes
  - New features (customer or market driven)
  - Competitive pressure
  - Refactoring (engineering but hard sell internally)
- Each release requires significant testing
- No hard rules, but:
  - 3-6 months minor version cycle
  - 12-24 months major version cycle
- Customers want/don't want new versions

Process

- Not enough process leads to chaos
- Too much is constipating
- The right amount depends on size and age
- Examples:
  - Process in software engineering
  - Process in decision making
  - Process in new initiatives
  - Process in personnel management

Product Management

- One of the hardest things – what is it?
  - Definition of products that will meet market needs, i.e. make money
  - Steering production thru product life cycle
  - Marketing understands what products are needed
  - Engineering understands what products it can build
- Tension between Engineering & Marketing
- Marketing must understand Engineering
  - But not micro-manage it

Part IV So you want to start a company?

Venture Capital Rule of Thumb:

Out of every 10 companies started:

- One makes the big time
- One more or less survives
- Eight fail

Realities of Funding

- This is 2006 – the bubble burst a long time ago
- To get significant funding:
  - Need more than ideas
  - Need a working product
  - Revenue stream almost a “must have”
  - Credible trials with good results
  - Well regarded, committed partners
- You don't own company – VC does
- Stock options were a carrot – but not anymore

How to Start

- Get people who know how to run a business
  - You don't
  - Good VC can help
  - You might be lucky – or you might not
  - Know your strengths and weaknesses
- Sales and Marketing critical
  - But very difficult
  - S&M won't be expert in product/technology
  - But you're not expert in S&M

Product Management (again)

- Really is critical
- Find product that market “needs”
- Not the same as one you can build
- Nor the same as having cool technology
- Need balance and respect between Engineering and Marketing:
  - Marketing shouldn't tell Engineering how to build it
  - Engineering shouldn't tell Marketing how to sell it

Innovation

- Very small "sweet spot" of innovation:
  - Too mainstream, no compelling value proposition
  - Too innovative, customers wary of engaging
- Standards important to customers
  - But investors want you to be different
- Great technology never enough:
  - Savings/ROI has to cover replacement
  - People cost: training, etc.

Intellectual Property

- What investors want most are patents
  - But patents are expensive
  - And often questionable (at best)
- Closed source route:
  - Customer only gets compiled code
- Or open source:
  - Can be protected thru appropriate licensing
  - Encourages “self marketing” of product
  - But a tough route with management and VC

Only Do One Thing

- You will never have enough people or time
- So have to do one thing – and do it right
  - Anything else is a distraction
  - When things don't work, tempting to have Plan B
- But plan B usually underestimated
- There are exceptions – but not many

Company Culture

- This is your chance to do it the way you want it
- Nice guys can win
- But nice guys can't be nice 100% of the time