Software Engineering
Coming of Age? Not Yet.

James Bach, Principal
Satisfice, Inc.
http://www.satisfice.com
james@satisfice.com

Coming of Age?
What does that mean?

- We’re finally figuring it out.
- We know what we’re doing.
- The major debates are over.
- Experts agree.
- Academia and industry are in sync.
- The good guys won.
I’m developing a hot product. My user base is potentially a million people.

I can’t afford much requirements work or in-house testing. But, I will use a lot of off-the-shelf parts and developers who I’m told are very good.

I’ll send the product out for beta testing.

I’ll collect feedback from a few hundred people selected from a friendly subset of potential users.

I’ll make whatever changes makes sense to me, based on beta feedback, then ship the product.

After that, I won’t change it for a few years.

Strengths…

– In-house testing and requirements costs very low.
– Won’t require in-house testing expertise.
– Beta testing focuses better on concerns of those users.
– Problems will be revealed that would not have been anticipated by in-house testers.
– No maintenance cost or risk.
– Arm’s length relationship to testers makes it easier to control the testing schedule.
– Open-ended quality standard helps us get to market more quickly.
**Analysis of the strategy...**

- **Weaknesses**
  - Off-the-shelf components may not work
  - Our developers may not possess the right skills or tools.
  - Beta tester selection is biased toward less critical users.
  - Beta tester response will be biased in favor of those testers who have more time.
  - We won’t know much about the quality of the actual testing that beta testers do.
  - Beta testers may not effectively communicate the problems they find.
  - We may not fully appreciate the problems reported.
  - The feedback loop to developers is windy and narrow.

**What about the cost of bugs curve?**

- Isn’t this QA plan a prime example of false economy? Not necessarily...
  - The famous “cost of bugs curve” only applies to defects that we choose to fix.
  - Failure costs in the field may be high, but we will not necessarily have to bear those costs.
  - Despite the risks, this plan may help us get to market quicker and achieve market share.
  - And if we fail? We’ll learn from that and move on.
My Opinion?

- Let’s not call this “engineering.”
- Let’s not pretend this will result in product we will *know* to be reliable and satisfying.
- Then again...

It *may* be a formula for getting rich in the software business by creating a “good enough” solution and creating barriers to competition.

What products are developed in this way?

- *The software engineering profession*
- The CMM
- The SWEBOK
- The ASQ CSQE Body of Knowledge
- Many standards
What’s immature about SE?

- Isolated enclaves
- Indoctrination without experimentation
- Experimentation without criticism
- Self-training or no training
- Dominated by short-term economics
- Perfunctory discourse
- Disinterest in related fields
- Unintelligible data
- Paradigm blindness

Do Our Students Study These Fields?

- Cognitive Psychology (*thinking and perception*)
- Family Psychology (*small group behavior*)
- Economics (*allocating scarce resources*)
- Graphic Design (*effective documentation*)
- Decision Theory (*rationality despite uncertainty*)
- General Systems Theory (*patterns in systems*)
- Qualitative Research (*observing behavior*)
- Measurement Theory (*limits of quantification*)

Who has the time???
During 4 additional weeks we were able to fix:
- 10 crash/data loss bugs
- 33 failures w/no workarounds
- 22 failures w/workarounds
- Not counting installation bugs

Contrasting Paradigms

<table>
<thead>
<tr>
<th>Objective</th>
<th>Cognitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>defined process</td>
<td>problem-solving</td>
</tr>
<tr>
<td>measurable progress</td>
<td>meaningful results</td>
</tr>
<tr>
<td>traceability to specs</td>
<td>relevance to mission</td>
</tr>
<tr>
<td>documented work</td>
<td>peer communication</td>
</tr>
<tr>
<td>repeatable work</td>
<td>effective work</td>
</tr>
<tr>
<td>organization focused</td>
<td>people focused</td>
</tr>
<tr>
<td>compliance praised</td>
<td>technical insight praised</td>
</tr>
</tbody>
</table>

Popular in Contract and Regulated communities
Popular in Market community
Paradigm Blindness

- **Objective:** process accountability matters
- **Cognitive:** solution accountability matters

What Should We Do?

- Oppose premature convergence on a BOK.
- Develop an ethnography of SE communities.
- Recognize the CMM as a cultural artifact of a particular sub-community of our field-- not natural law.
- More fieldwork.
- Study and catalog software project dynamics.
- Study *naturally situated cognition* in software teams.
- Study exploratory and heuristic process definition.
- Study the arts of conferring.
- Bring some critical rationalism into our field.
Conferring

- The LAWST conference
- Consultants Camp
- Jerry Weinberg’s Courses and Communities
  - www.geraldmweinberg.com
- Context-Driven Testing Forum
  - www.egroups.com/groups/software-testing
- The AYE conference
  - www.ayeconference.com

Teenage Psychology

- Teenagers seek the appearance of maturity, because maturity means power and identity.
- Teenagers are impatient with the work it takes to become mature.
- Teenagers are often oblivious to the many advantages of youth.
- Adults know that with maturity comes many onerous burdens.

We are a young and confused field.
Make the most of it.