The Social Side of Software Engineering

Clay Williams, PhD
IBM Thomas J. Watson Research Center
Agenda

- A Brief History of Software Engineering
- The Situation Today
- Five Weird Elements of Software Engineering
- The Emergence of Social Concerns
- Research Challenges and Conclusion
A Brief History of Software Engineering

1968: NATO Science Committee Meeting
Dijkstra: Goto Considered Harmful

1972: Parnas: Decomposition and Information Hiding
Dahl and Hoare: Hierarchical Program Structures

1974: Mythical Man Month
Structured Design

1975: Structured Analysis
JSP

1977: Z
Structured Calculus
ERM

1987: No Silver Bullet
VDM
JSD

1997: UML
Design Patterns
Separation of Concerns
Product Lines

2010: Socio Technical Congruence
Agile Manifesto
Where are we today?
Maintenance Costs Predominate

Percentage of IT Budget Devoted to Maintenance

![Bar chart showing the percentage of IT budget devoted to maintenance over different decades.](Moad (Gartner) 1990)
Money for New Development Scarce

Expanding maintenance costs are forcing a reduction in development costs.
Yet Software is Everywhere

60s / 70s
- Back office
- Payroll
- Accounting
- HR

80s
- Front office
- Spreadsheets
- Word Processing
- Internal Communications

90s and beyond
- WWW
- Supply Chain
- ERP
- CRM
And We Still Struggle with Outcomes

Standish Group - CHAOS Report

<table>
<thead>
<tr>
<th>Year</th>
<th>Success</th>
<th>Failure</th>
<th>Challenged</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1998</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Five Weird Elements of Software Engineering
I: Stakeholders and Requirements
2: Problem to Solution Mapping

- Interaction Paradigms

- Logic

- Data

Plus the non-functional properties:
- Performance
- Usability
- Maintainability
- Scalability
- Security
- …
3: Coding, Compilation, and Execution

“Linearization”

Compilation

Execution And Debugging
4: Malleability
5: Doing this all in Teams
Please stop comparing creating software to building bridges and buildings (or even hardware)!
The New Primacy of Social Concerns
The Primary Concern of 21st Century Software Engineering: Sustainable Development

- Development that meets the needs of the present without compromising the ability to meet future needs.

More Detail...

- **Value Today:**
  - Building the right features using the right skills
  - On a reasonable schedule
  - At a reasonable cost
  - With sufficient quality that problems don’t break the bank
  - With sufficient adaptive capacity to respond to future needs
  - Using a reasonable pace of work

- **Value Tomorrow:**
  - Adapting to changing demands, while managing schedule, cost, quality, flexibility, and sanity
Right Features

• Requirements Engineering - inappropriate focus on requirements representation

• Agile Corrective - simple user stories

• Social factor: high, focusing on communication with stakeholders and team

• Technical Challenges - story prioritization and selection; optimization across programs
Reasonable Schedule and Cost

- Estimation Research - inappropriate focus on parametric models
- Agile corrective - simple estimates; timeboxing; sophisticated analytics
- Social factor: high, focusing on understanding perspective of people doing the work, as well as implications
- Technical Challenges: combination with parametric models in early lifecycle of large projects

Healthy programs work off risk early and track close to the curves
Reasonable Quality

- Testing Research - irrelevant from above (model checking) and below (monkey testing)
- Agile Corrective - test first design; test cases as specification
- Social Factor: high, test cases as a basis for communication rather than correctness only
- Research areas: test case adequacy for test first methods
Adaptive Capacity

- Architectural Work: too much focus on representation and semantics
- Corrective: Applying options and socio-technical thinking to software architecture
- Social Factor: high, architecture as a basis for team coordination and flexibility
Reasonable Pace of Work

- Current Practice: the long-working, long-suffering hero
- Agile Corrective: Timeboxing, extensive work tracking, intentional pacing
- Social factor: high, work items as basis for both work and awareness
- Research topics: understanding actual time spent on work items. Traceability to other artifacts. Others.
Current Research and Challenges
What is IBM up to?

Visit:
http://jazz.net and
http://open-services.net

**Rational Team Concert**
A collaborative work environment for developers, architects and project managers with unified work item, source control and build management, process management and iteration planning.

**Rational Quality Manager**
A web-based test management environment for quality professionals that provides a customizable solution for test planning, workflow control, tracking and reporting.

**Rational Requirements Composer**
A requirements definition solution with easy-to-use elicitation and definition capabilities that enable the refinement of business needs into unambiguous requirements.

**Rational Project Conductor**
A project and resource management system optimized for software and systems delivery, enabling project and program managers to plan, schedule, and staff projects, with the right resources working on the right tasks.
What is IBM Research up to?

Business Stakeholders

Operationa Stakeholders

Delivery Stakeholders

Dossiers: Actionable, Role, and Context Specific Presentation

Crescendo: Connecting Enterprises Delivering Software
- Transparency and Awareness
- Process Support
- Collaboration
- Organizational Structure

AnODe: Analytics and Optimization for Development
- Simulation
- Forecasting
- Prediction
- Optimization

Development Data

Features
What are other institutions up to?

- CMU: Socio-Technical Congruence
- Harvard Business School: Design and Optionality
- Virginia: Cyber-Social Systems; Adaptive Capacity of Architecture
- UC-Irvine: Coordination and collaboration in globally distributed teams
- UC-Davis: Defect bias; Social connections in OSS
- UBC: Coordination and requirements
- Drexel: Socio-Technical Coordination; Adaptation
- Nebraska: Visualizing and Exploring Socio-Technical Coordination
- ...
Conclusion

- Challenges of building software remain
- Social thinking forms a new basis for software engineering
- Technical challenges remain, but must be solved with social context in mind
- Validating this kind of research is *really hard*
- Many challenges and opportunities ahead!