Quality & Evolution: some relationships

Michel Wermelinger
Computing Department
The Open University, UK
http://michel.wermelinger.ws
Product Evolution

Bennett & Rajlich. Software maintenance and evolution: a roadmap. ICSE 2000
 Principle of Software Uncertainty

- The real-world outcome of any E-type software execution is inherently uncertain with the precise area of uncertainty also not knowable

Declining Quality

- Unless rigorously adapted and evolved to take into account changes in the operational environment, the quality of an E-type system will appear to be declining
  - 7th law of software evolution
Continuing change

- An E-type system must be continually adapted or else it becomes progressively less satisfactory in use
  - 1st law of software evolution
Continuing growth

- The functional capability of E-type systems must be continually enhanced to maintain user satisfaction over the system lifetime
  - 6th law of software evolution
Punctuated Equilibrium

Wermelinger et al. Design principles in architectural evolution: a case study. ICSM 2008
Increasing complexity

- As an E-type system is changed its complexity increases and becomes more difficult to evolve unless work is done to maintain or reduce the complexity
  - 2\textsuperscript{nd} law of software evolution

![Diagram showing stages of software evolution: perfective, adaptive, corrective, preventive]
Maintenance

<table>
<thead>
<tr>
<th></th>
<th>Correction</th>
<th>Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proactive</td>
<td>Preventive</td>
<td>Perfective</td>
</tr>
<tr>
<td>Reactive</td>
<td>Corrective</td>
<td>Adaptive</td>
</tr>
</tbody>
</table>

April. Studying Supply and Demand of Maintenance Services. QUATIC 2010
Maintainability

Luijter & Visser. Faster defect resolution with higher quality of software. SQM 2010
Stable Dependencies Principle

- Dependencies should be in the direction of stability


Wermelinger et al. Design principles in architectural evolution: a case study. ICSM 2008
Open/Closed Principle

- Entities should be open for extension but closed for modification.

Meyer. Object-Oriented Software Construction. Prentice Hall 1988
Cloning considered harmful

Maintainability of clones

Lozano & Wermelinger. Assessing the effect of clones on changeability. ICSM 2008
Evolution of quality cost

Evolution for quality prediction

Nagappan et al. Change Bursts as Defect Predictors. ISSRE 2010
Process evolution

- Processes and applications are both executed, they both address requirements that need to be understood, both benefit from being modeled by a variety of sorts of models, **both must evolve guided by measurement**, and so forth.

Osterweil. Software processes are software too, revisited. ICSE 1997
Process Improvement

CMMI Staged Maturity Levels

- Level 1: Initial (Process unpredictable, poorly controlled and reactive)
- Level 2: Managed (Process characterized for projects and is managed)
- Level 3: Defined (Process characterized for the organization and is proactive)
- Level 4: Quantitatively Managed (Process quantitatively measured and controlled)
- Level 5: Optimizing (Focus on continuous process improvement)

from cmmilevels.com
Quality management

Plösch et al. A method for continuous code quality management. QUATIC 2010
Bettin et al. A PMO Installation for IT Project Management. QUATIC 2010
Education

- Holistic view
  - Human / Social
  - Legal / Ethical
  - Economic
Conclusions

- E-type evolution processes are multi-level, multi-loop, multi-agent feedback systems
  - 8th law of software evolution