Industrial perspective on Software Evolution

Joost Visser
Company

- Spin-off from CWI in 2000, self-owned, independent
- Management consultancy grounded in source code analysis
- Innovative, strong academic background, award-winning, profitable

Services

- Software Risk Assessments (snapshot) and Software Monitoring (continuous)
- Toolset enables to analyze source code in an automated manner
- Experienced staff transforms analysis data into recommendations
- We analyze over 50 systems annually
- Focus on technical quality, primarily maintainability / evolvability
Who is using our services?

- Financial
  - ABN-AMRO
  - ING
  - Rabobank
  - Friesland Bank
  - LeasePlan
  - Interpolis

- Public
  - PGGM
  - Zorg en Zekerheid
  - InterBank
  - ZwitserLeven
  - delta Lloyd
  - Allianz
  - ProRail
  - Beheer ZoekerWoning

- Logistics
  - KLM
  - Getronics
  - PinkRocade
  - DHL
  - RDW
  - TNT
  - EUROMAXX
  - Norfolkline
  - Belastingdienst

- IT
  - KPMG
  - Capgemini
  - PriceWaterhouseCoopers
  - essent
  - ENECO

- Other
  - CENTRIC
  - Alcatel-Lucent
  - CHES
  - AS"
Why change?

In modern business

“Change Is The Constant”

Drivers of change

- New or improved products and services to innovate, to compete
- Reduction of operational costs to be profitable
- Laws and regulations to comply
- Adapt to external change for continuity
- Fix bugs introduced due to the above
What is so difficult?

**Obstacles to change**

- You can’t contract innovation. Contractors need stable specs.
- Limited anticipation. Future needs are unpredictable.
- Synchronization. Interconnected systems, artifacts, people, organizations must change together.
- Change reduces changeability. Lehman.

+ Change

- Complexity

- New can not ignore old. Legacy.
- Knowledge dissipation. You can’t change what you don’t understand.
Functional vs technical quality

Software with high technical quality can evolve with low cost and risk to keep meeting functional and non-functional requirements.
Example: “standard” packages

Standard package
- E.g. SAP, Siebel, Sharepoint, …
- Need “configuration” and “customization”

Promise
- Quick - up and running in no-time
- Universal - lots of pre-existing functionality, extensible (by third parties)

Issues
- Vendor lock-in by the overall product
- Vendor management issues regarding the many third-party contributors
- Change becomes increasingly slow and increasingly costly
  E.g. “upgrade problem” results from vendor lock-in + sheer complexity of many third-party plug-ins and extensions

Challenges
- Evolving these systems away from vendor lock-in
- Managing the (changing) interfaces towards services of third parties
Example: service orientation

**Service orientation**
- Make existing resources available as services

**Promise**
- Simple - integration without clutter
- Dynamic - plug, unplug, recombine services at will

**Issues**
- Pervasive (vendor-specific) SOA technology
- Misunderstanding of “service” concept
- Requires business case at the level of the entire organization

**Challenges**
- When, why, and how introduce service-orientation?
- Avoid technology-trap
- How to co-evolve services and service architecture
Broader perspective

**One hand**
- Strong desire for quick, embracing solutions
- Concrete benefits, known costs

**Other hand**
- Longer term consequences are unquantified, invisible …
- … therefore not *actionable*

**Challenge**
- Make *evolvability* measurable, visible, actionable

**Requires**
- Broad empirical research program
- Sharing of data, trans-organisational analysis
Dr ir Joost Visser

E  j.visser@sig.nl
W  www.sig.nl
T  +31-20-314-0950
M  +31-61-448-8847