Practical API Design

- **Jaroslav Tulach**
-

Sun Microsystems
http://www.netbeans.org

NetBeans

Agenda

What makes good API?
Like a free speech
Examples
Q&A

What makes an API good?

- API is used for communication
  - build trust, clearly describe plans
- Must be:
  - a) beautiful
  - b) correct
  - c) simple
  - d) good performance
- Not necessarily!

What makes a technology good?

- Coolness
  - > before adoption
- Time to market
  - > at beginning of adoption
- Total Cost of Ownership
  - > then and forever

API is a technology facade

- Hype & Community
- Limited understanding
- Empiristic programming
  - > Cluelessness
- Error forgiving
- Easy upgradability

Rules for Successful API design

- Use case driven API design
  - use cases -> scenarios -> javadoc
- Consistent API design
  - An interface that is predictable serves better than one which is locally optimal but inconsistent across the whole set.
- Simple and clean API design
  - less is more - expose only necessary functionality
- Think about future evolution
  - First version is not going to be perfect

Like Human Rights

What is the most important right?

The right for free speech. Because if you have right to speak freely, you can always talk about the missing ones and ask for them.

Karel Havlíček Borovský

Evolution

- First version is insufficient
- Incremental changes
- Knowing your clients is not possible
- Preservation of Investments
- Agile API Design

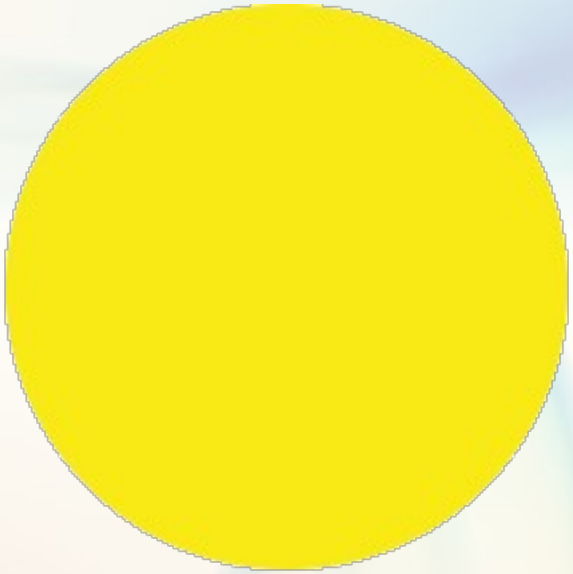
Backward Compatibility

- Source
  - > ability to compile against new version
- Binary
  - > ability to link to new version
- Functional
  - > ability to compute the same result

Evaluation of an API Quality

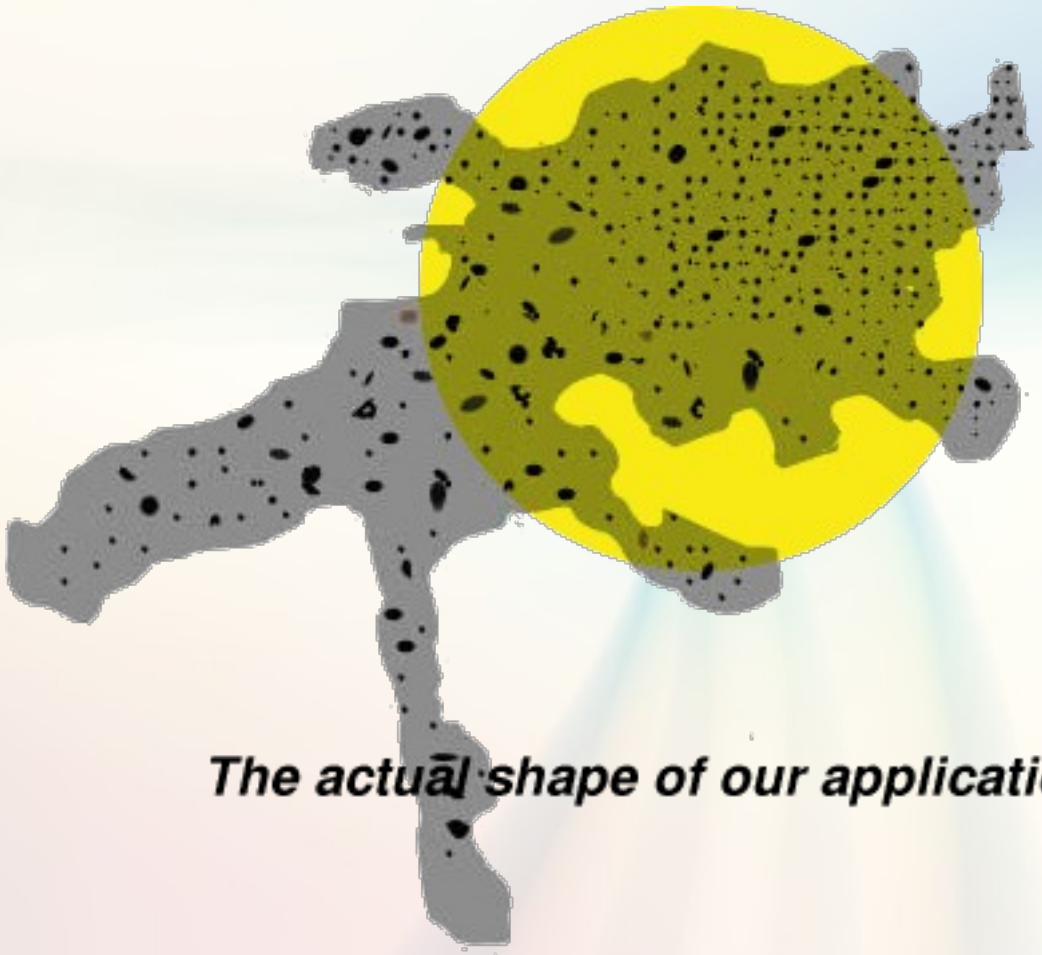
- Customer centric – easy to use
- Use cases, scenarios, javadoc
- Future evolution
- Test coverage
- quality = code Δ specification
- the "amoeba" model

The Amoeba Model



How we think our application looks like

The Amoeba Model



The actual shape of our application

The Amoeba Model

Shape of amoeba after next release

API Fest

- Agile to ad-absurdum
- Only about evolution
- Not a beauty contest
- Evaluated by you

API Fest '08 Roadmap

- Solutions for Task 1 arrived
  - need to align
- Additional 2-5 rounds
  - once, twice a week
- Judgment Day
- Finish by Oct 23, 2008

API Fest Don'ts

- Do not fix previous mistakes
  - usually leads to incompatibilities
- Runtime aspects are important
- Do not remove anything from the API
- Add carefully
  - new class/interface is OK
  - new method in subclassable type is not OK

API Fest Gotchas

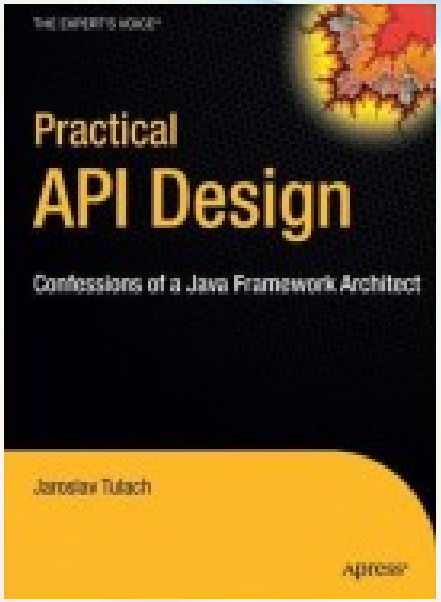
- Seeking for evolution problems with God's eyes
  - knowing all the previous versions
  - line by line diffs
- No permissions, please
  - no reflection
  - no getClass().getName()

Questions & Answers

Practical API Design

Confessions of a Java Framework Architect

ISBN-10: 1430209739



Jaroslav Tulach
http://www.apidesign.org