Architecture War Stories

Fun Stories from an Architect’s Life

Stefan Tilkov | @stilkov | innoQ

GOTO Berlin
11/07/2014
Ground Rules

No offense – just lessons learned

Most systems are in production

Still, many {managers | users | admins} are happy
Less can be more
Context

Customer care, accounting & billing

Large-scale big-bang replacement effort

› 150 developers

› 20 architects/framework devs

Java, Swing, CORBA, Oracle

Custom persistence framework
Global synchronization? 2 PC?

Cache notifications? Events?

Transaction locality? Client Affinity?
Lessons learned

Don’t cache the cache

In distributed systems, application state is your enemy

If it’s “sophisticated”, you probably shouldn’t be doing it

Development ≠ production
Data Complications
Context

Insurance system

Maintenance 1000 person months/year

Smalltalk, COBOL, C, IMS, DB2

Homegrown COBOL OO system
Migration
Lessons learned

Data should be free of code dependencies

The longer you store it, the simpler it needs to be

Beware of smelly blobs
Evaluating Yourself to Death
Context

Strategic health care platform

Architecture team 20+10 people

Build foundation for 800+ person year effort
Vendor Selection

Week 0  Collect and agree on requirements

Week 8  Conduct market research

Week 10  Send out RFP to selected vendors

Week 14  Evaluate responses, create shortlist, start PoC

Week 20  Evaluate PoC results, recommend vendor X

Week 26  Accept your CEO picked vendor Y
Vendor Selection

Week 0: Collect and agree on requirements

Week 8: Conduct market research

Week 10: Send out RFP to selected vendors

Week 14: Evaluate responses, create shortlist, start PoC

Week 20: Evaluate PoC results, recommend vendor X

Week 26: Shutdown company
Lessons learned

Just do it

Better ask for forgiveness than permission
Your system WILL be dynamic
Context

Large-scale insurance system

Model-driven development

> 100 developers

2 Releases/year
What if you miss your slot?
Lessons learned

Centralized responsibility hurts

Faced with too much rigidity, a way around the rules will be found

Just because you’re used to it doesn’t mean it’s acceptable
Formalism and Common Sense as Orthogonal Concepts
Context

Telecommunications OSS platform

Support new devices without code changes

› 100 developers

› 3 countries
Warning

((Meta) meta) models) are a perfect means to waste time, on a highly inspiring intellectual level, without any results for years

Highly complex methodology, tool support, agreement processes

Consultants with exclusive pricing strategies
Lessons learned

If it makes you want to scratch your eyes out, you might not want to do it

Sometimes constraints need to be fought

Beware all things “meta”
The Perils of Being Smart
Context

Complex enterprise-wide SOA effort

Custom-built “lightweight” middleware
Consumer/Provider

Business Logic

Messaging API

Middleware Lib

Transport Abstraction

Configuration Service

Registry Service
Lessons learned

Fight the madness

Tunneling is the root of all evil

Don’t be too smart
Non-extensible Extensibility
Context

E-Commerce (retail) provider

Global customer base

Catalog/CMS/Shop/Fulfillment

Multi-tenant

Highly customizable
Large customers
("strategic")

Small customers
("long tail")

The Solution

(Acceptable)
Cost

High

Low

General

Specific

Customization
( Need )
Lessons learned

Don’t be all things to all people

Sometimes specific code is preferable

Saying “no” might help
Summary
Never-ending supply of
{war stories | catastrophes | disasters}
seems guaranteed
Advice:
1. Give Feedback
2. Reflect
3. Iterate
Q&A

Stefan Tilkov, @stilkov
stefan.tilkov@innoq.com
Phone: +49 170 471 2625