coding {the} architecture

Master builder

The original generalising specialist?

@simonbrown
#gotoams

simon.brown@codingthearchitecture.com

@simonbrown on Twitter

Writing

Training and consulting

Published incrementally

Variable pricing

Buy now and get free updates

Leanpub

leanpub.com

goto; conference

In this track we relate the origins of the architect back to its roots of being a master-builder and ask why our industry has moved away from this in the real world.

You will hear some real stories from master-builders as we take a look at how they approached the software architectures that they created.

Big up front design

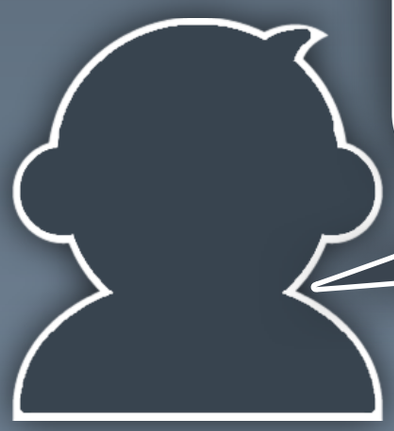
and analysis paralysis

Waterfall

UML

I'm a

software architect



Ivory Tower

PowerPoint Architect

Architecture Astronaut

Software development
is like
${some.metaphor}

`${some.metaphor}`

is broken because...

:-p

“architect”

Latin: architectus

Greek: arkhitekton

“master builder, director of works”

arkhi- "chief"

+

tekton "builder, carpenter"

The Conclusion

Building architects were master builders,

therefore

software architects should code
rather than sitting in ivory towers

The End

Back to the Middle Ages

Master builder = “master mason”

a stonemason

most major buildings were constructed of stone at the time

A master mason, is a

manipulator of stone,

an artist in stone and

a designer in stone.

Did master builders actually build?

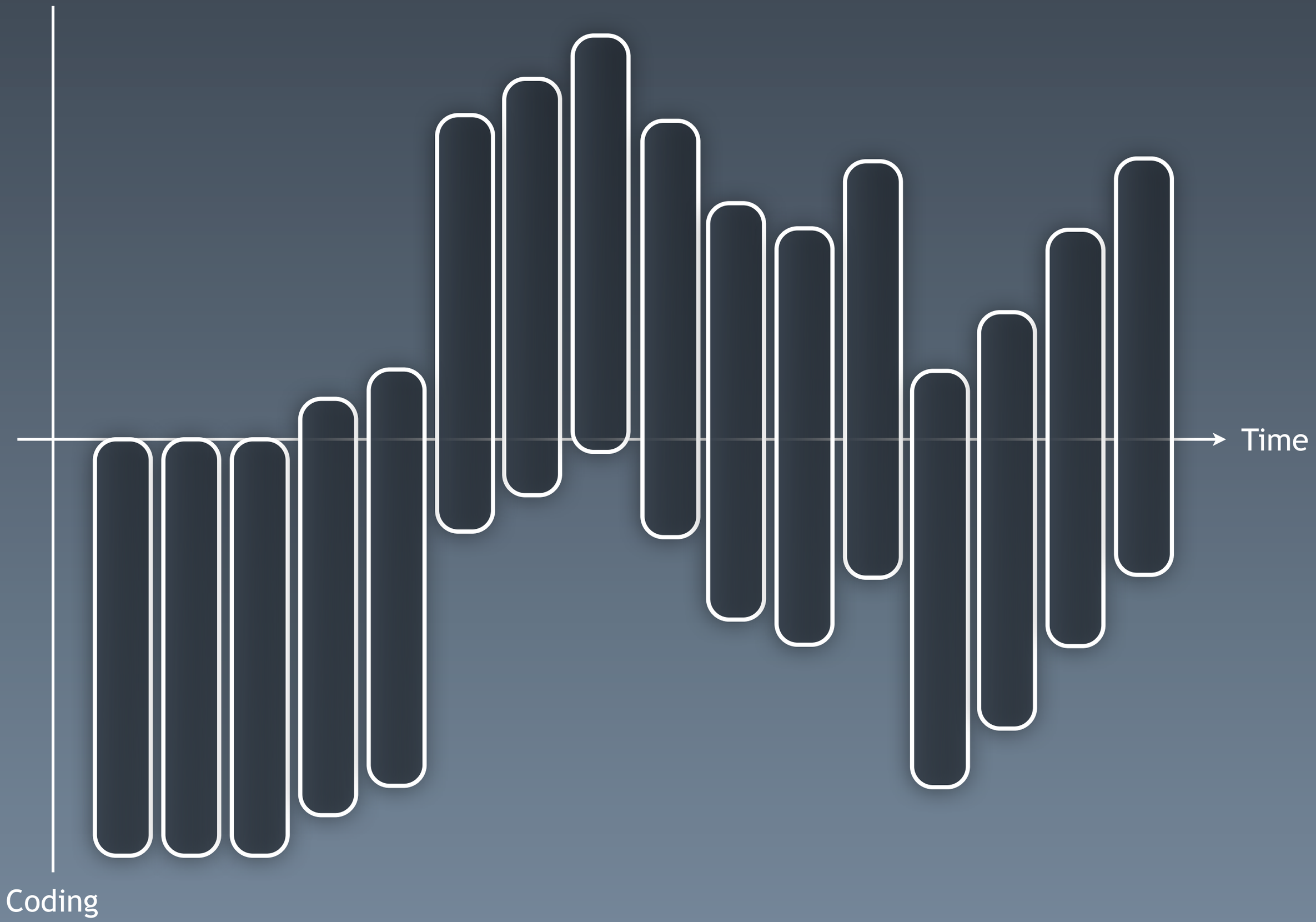
Although a master mason was a respected and usually wealthy individual, he first had to

prove his worth by going through the ranks as a stonemason

and then a supervisor, before being appointed to the highest position in his trade.

Throughout ancient and medieval history, most architectural design and construction was carried out by artisans, such as stone masons and carpenters,

rising to the role of master builder.


“Architecture”
Time
Coding

How much contact he actually had with this substance is, however, debatable.

Ivory towers?

A mason who was at the top of his trade was a master mason. However, a Master Mason, by title, was the man who had overall charge of a building site and master masons would work under this person. A Master Mason also had charge over carpenters, glaziers etc. In fact,

everybody who worked on a building site was under the supervision of the Master Mason.

The master mason, then, designed the structural, aesthetic and symbolic features of what was to be built; organised the logistics that supported the works; and, moreover, prioritised and decided the order of the work.

...

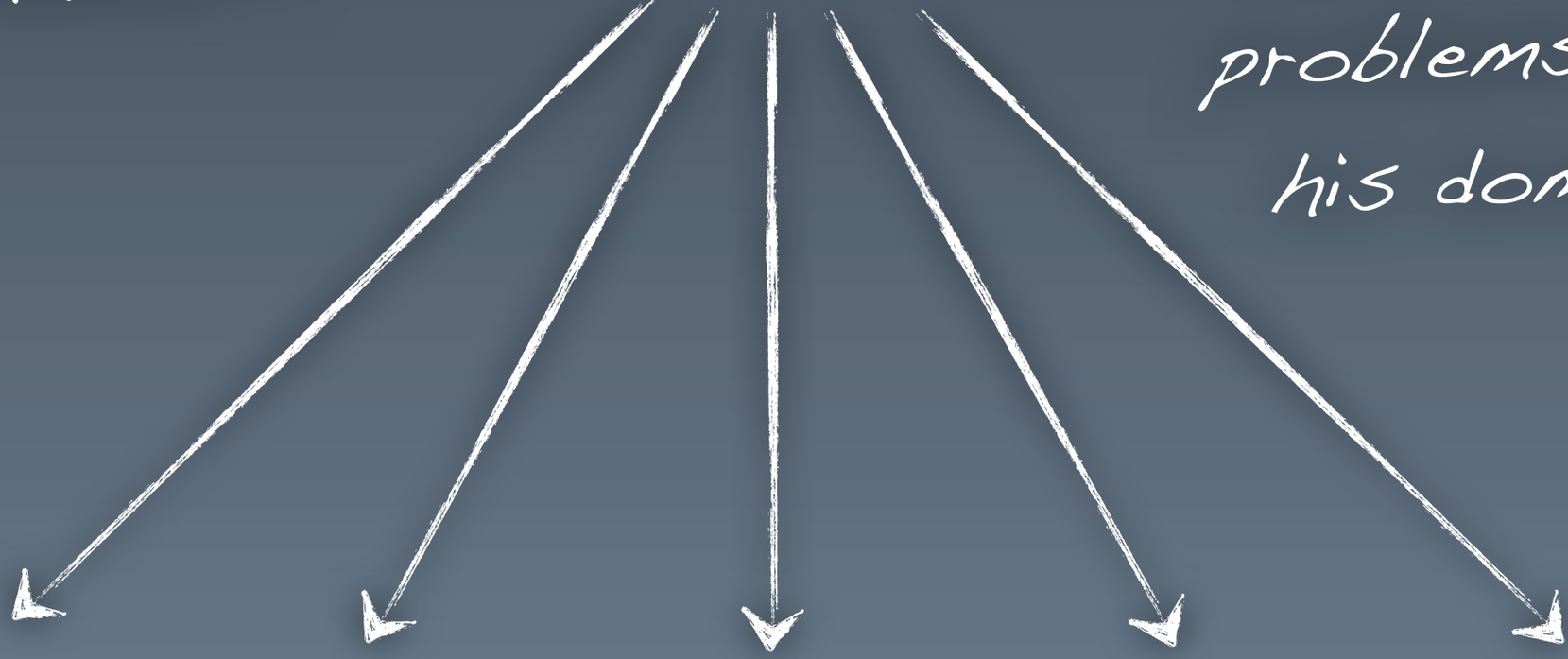
If, as seems likely, this multiplicity of tasks was normal it is hardly surprising that master masons took little part in the physical work

(even had their status permitted it).

" Every lesser mason followed the directions set by the master...

Master Mason

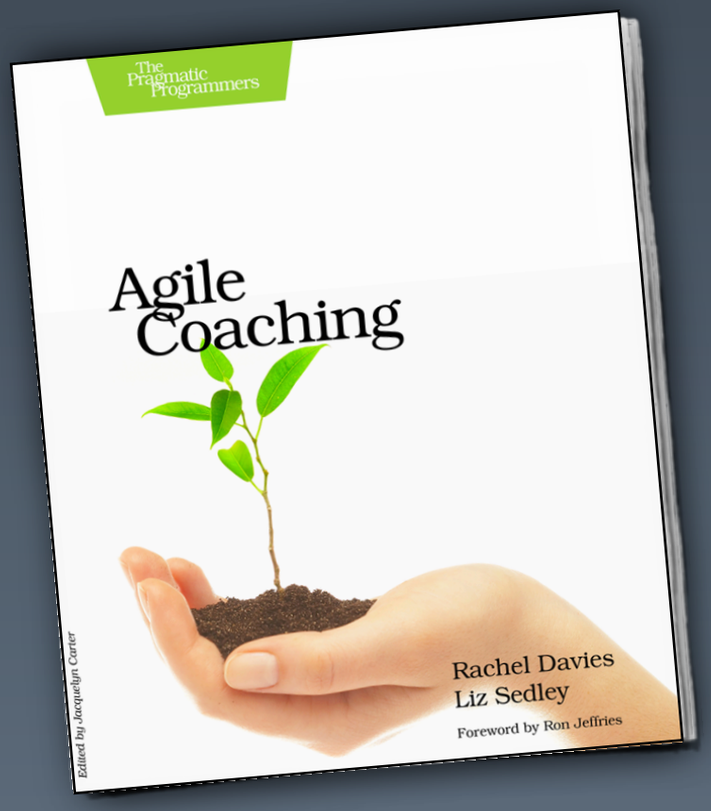
... and all decisions with regard to major structural, or aesthetic, problems were his domain."



Mason Mason Mason Mason Mason

Testimony of this supposition is supplied by a sermon given in 1261 by Nicholas de Biard

railing against the apparent sloth of the master mason 'who ordains by word alone'.



Agile Coaching

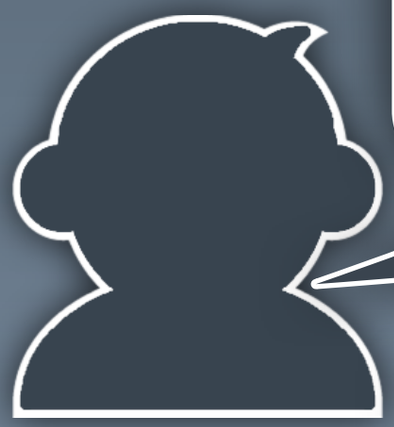
Rachel Davies & Liz Sedley

"If you know how to program, it's often tempting to make suggestions about how developers should write the code. Be careful, because you may be wasting your time—

developers are likely to ignore your coding experience if you're not programming on the project.

They may also think that you're overstepping your role and interfering in how they do their job, so give such advice sparingly."

Why do you want to `code` on this project?

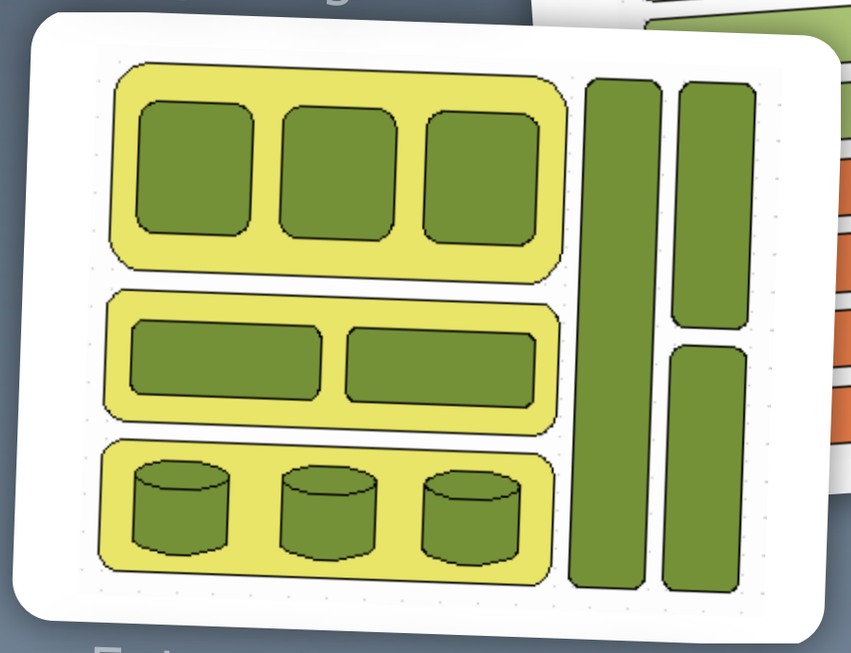


Manager

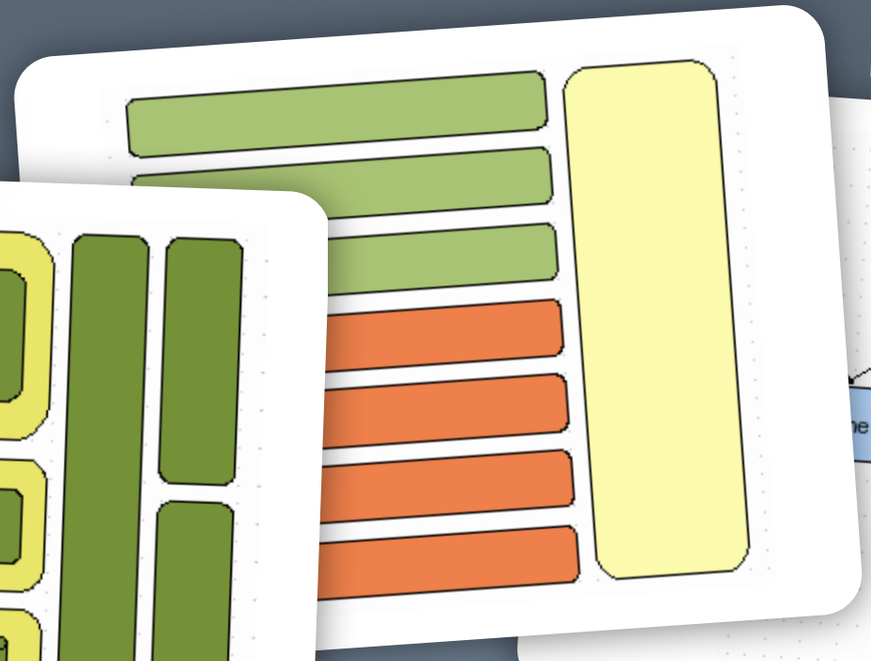
PowerPoint

Architecture

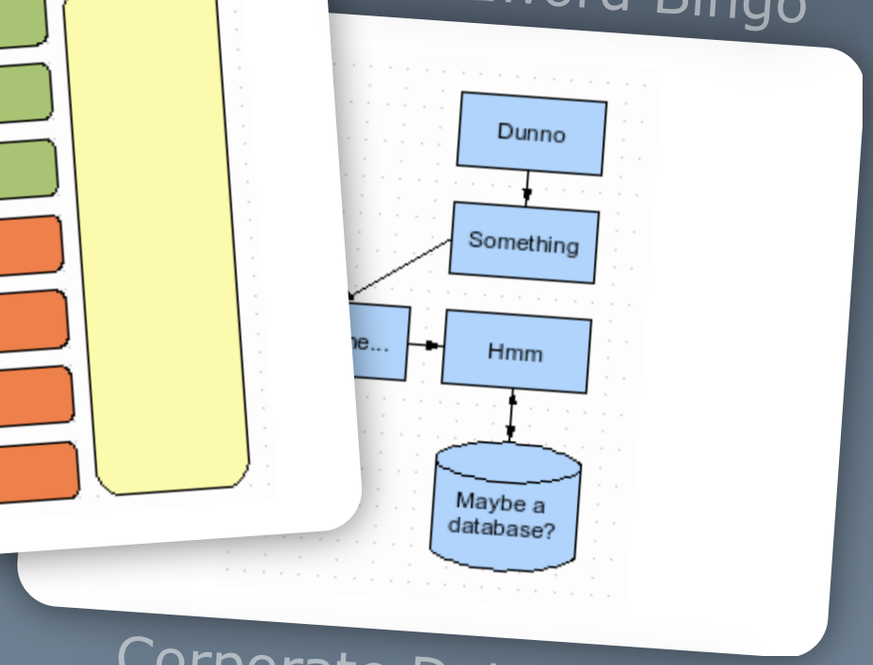
Single Sign-On



Enterprise Service Bus

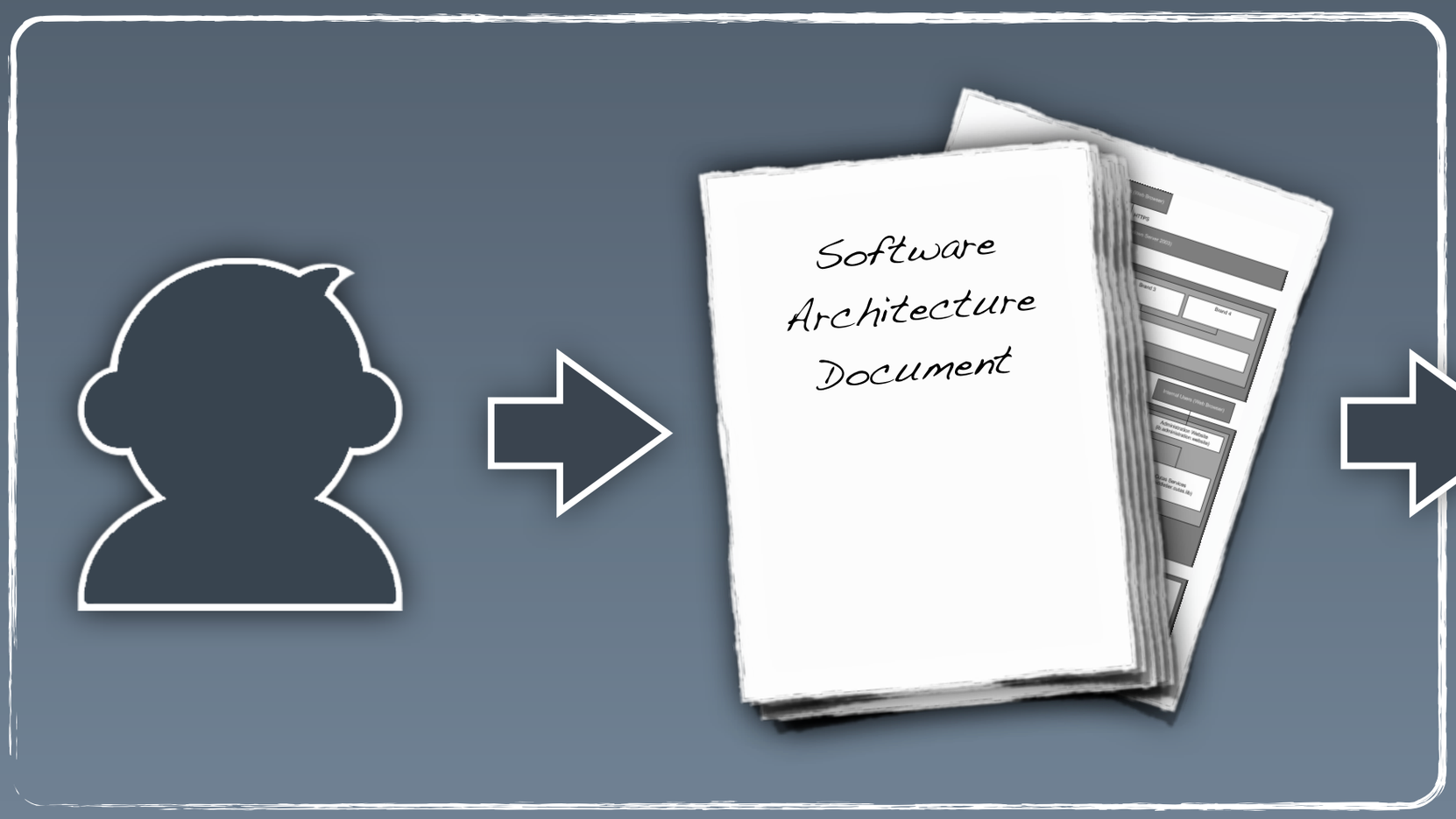


Buzzword Bingo



Corporate Data Model

Software development is not a relay sport



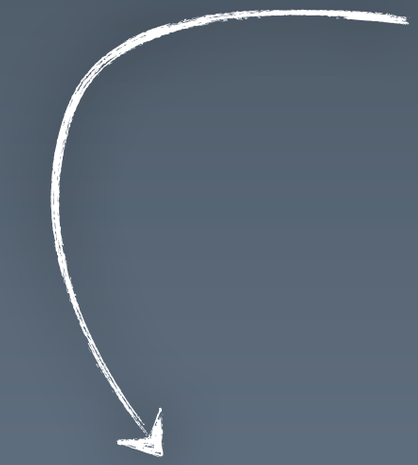
AaaS ... architecture as a service

Successful software delivery

is **not** an

implementation detail!

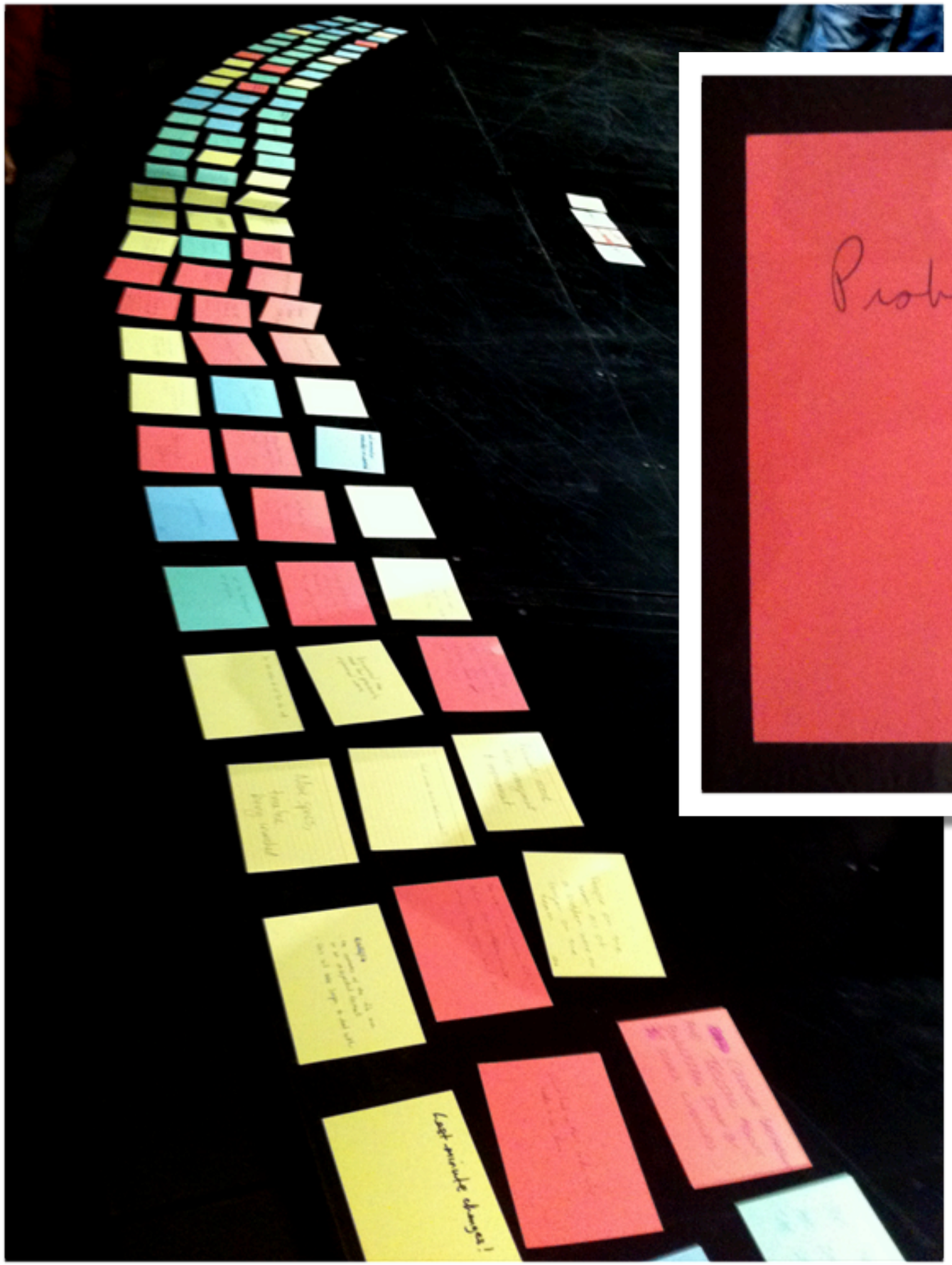
Why did you choose technology A?

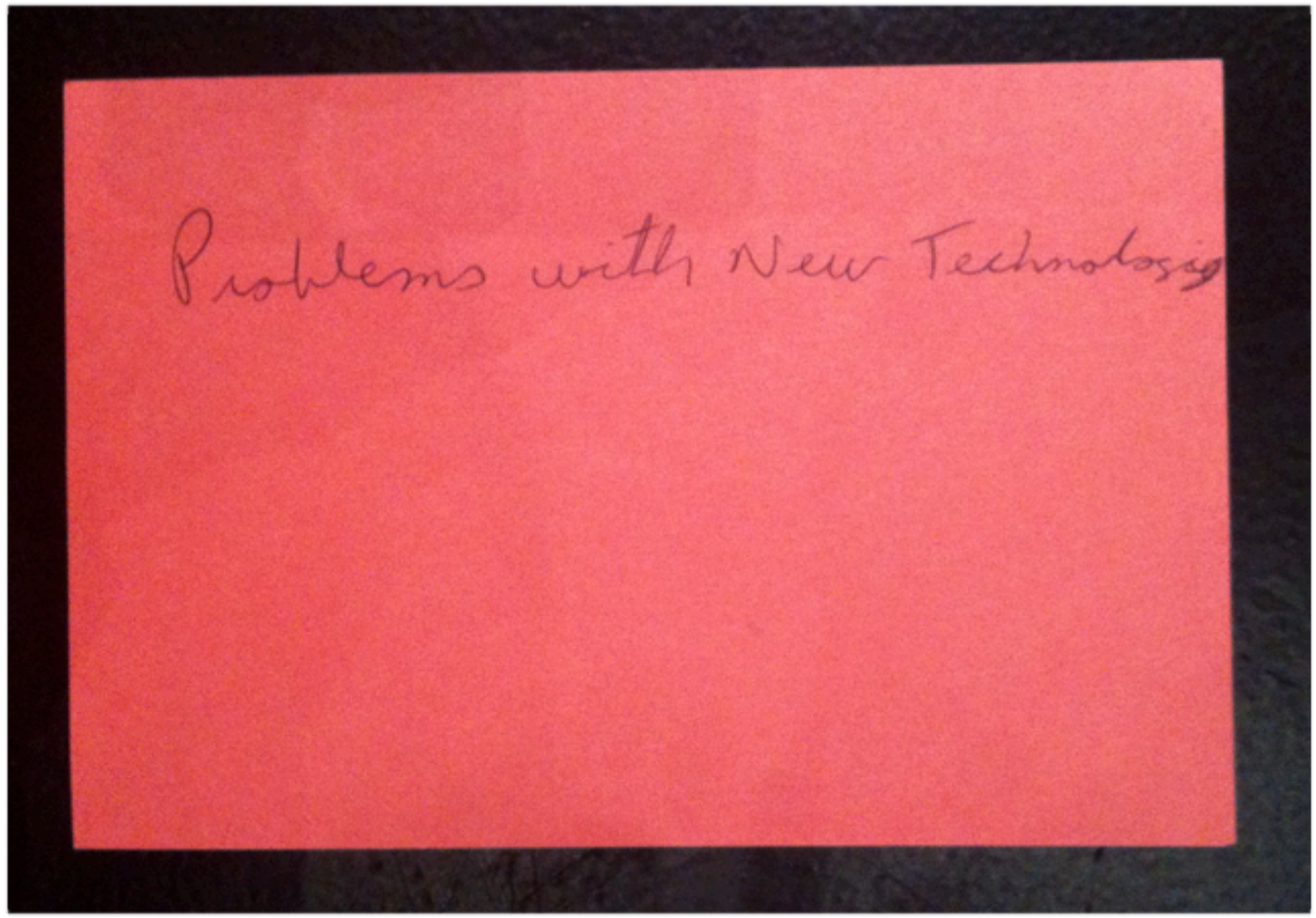


A

B

C





An example timeline from

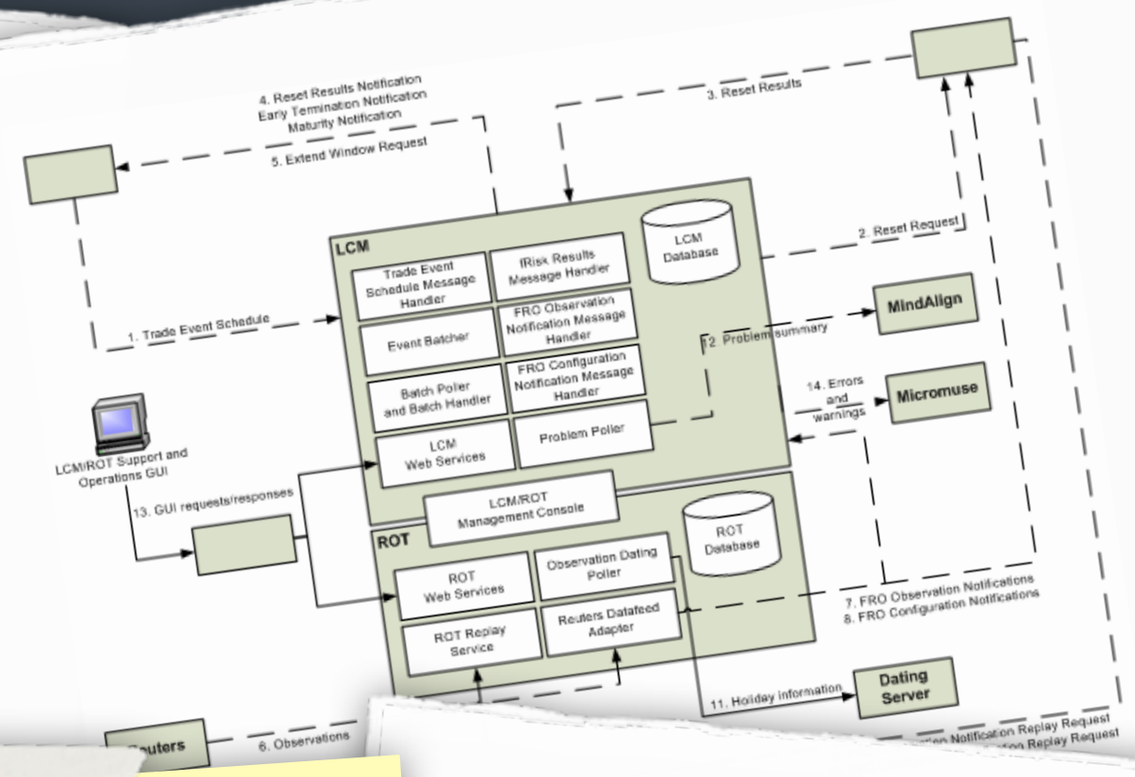
“Beyond Retrospectives”

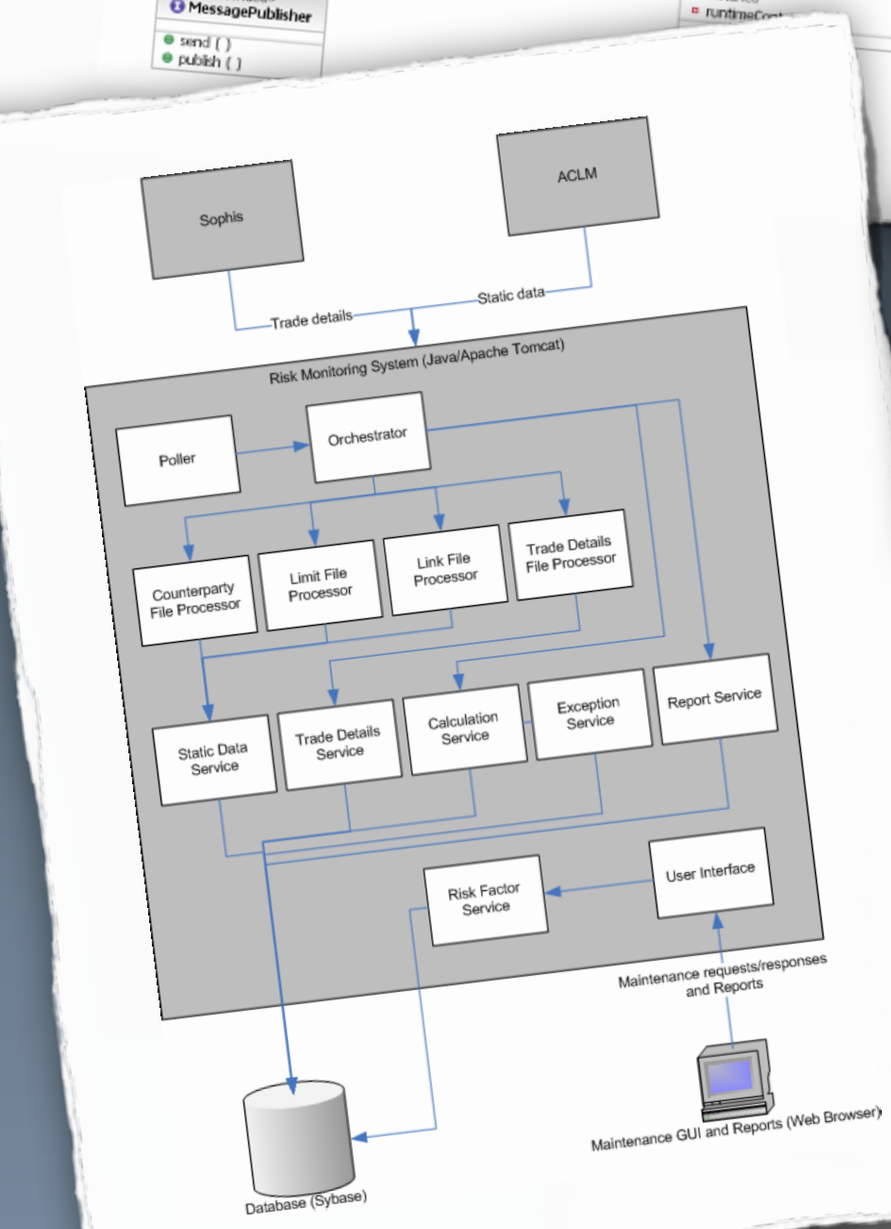
by Linda Rising

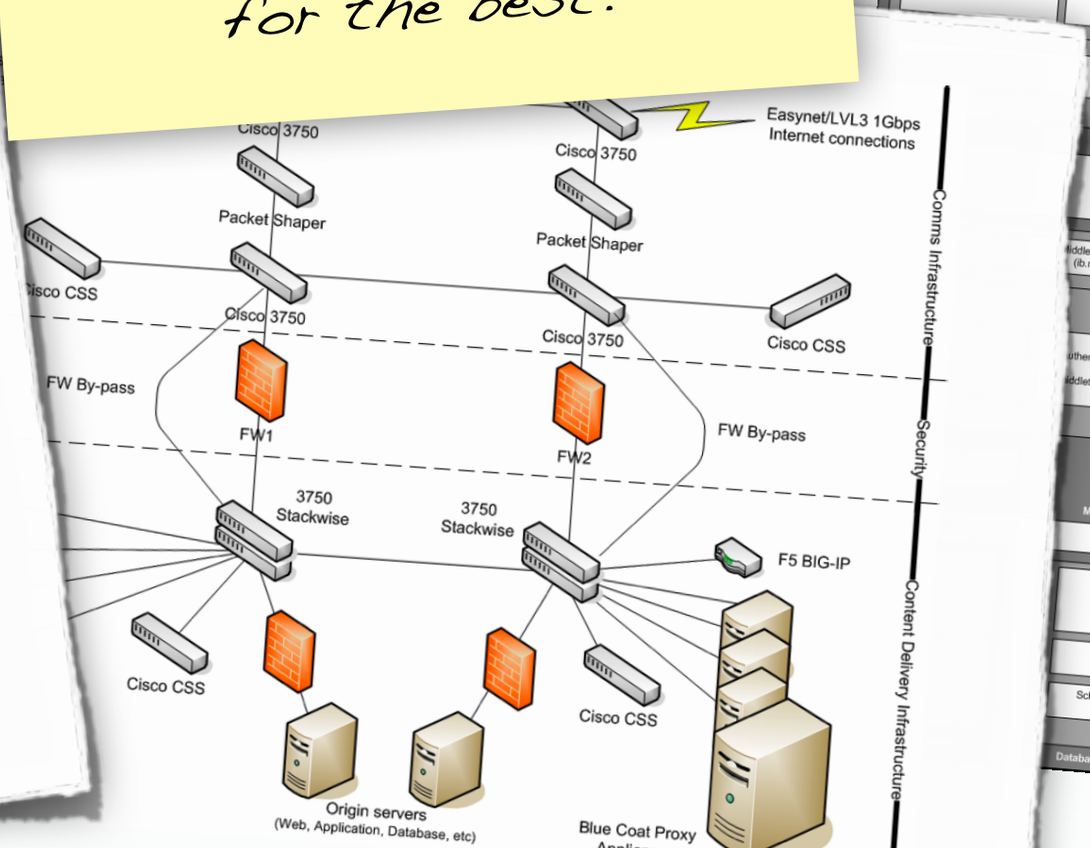
#gotocon Aarhus 2011

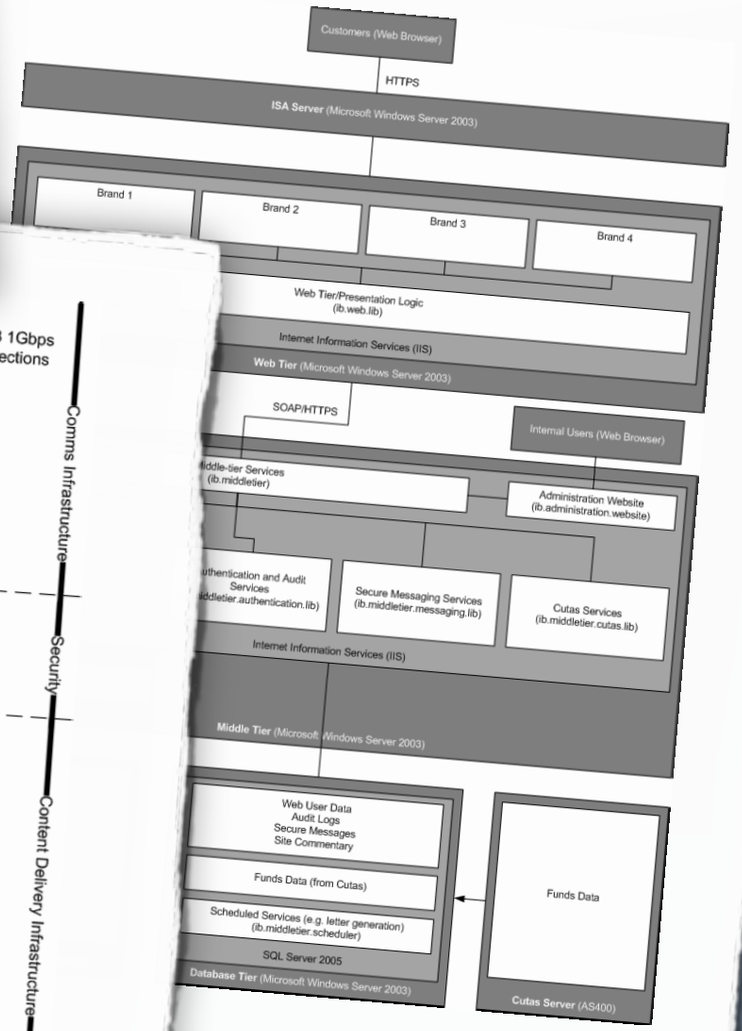
Foolishly hoping

for the best?

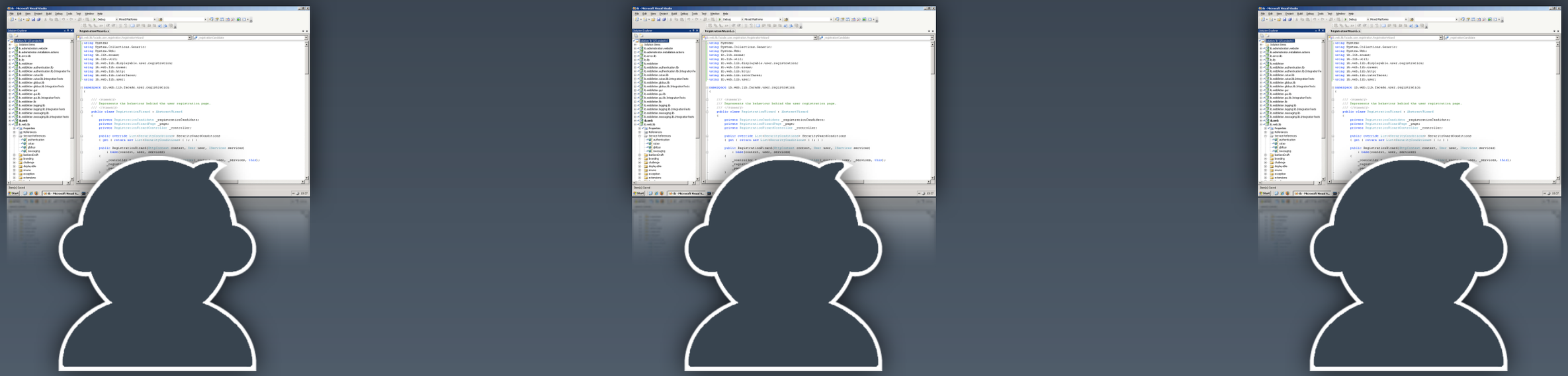






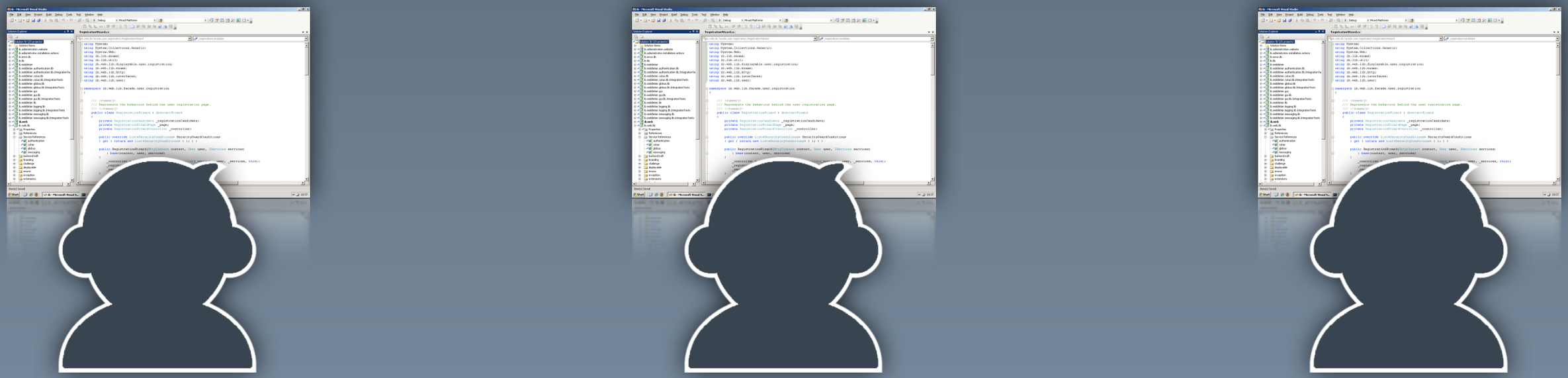


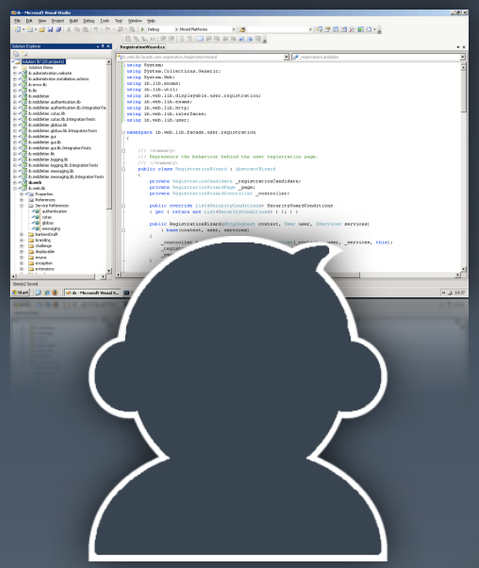
Leadership

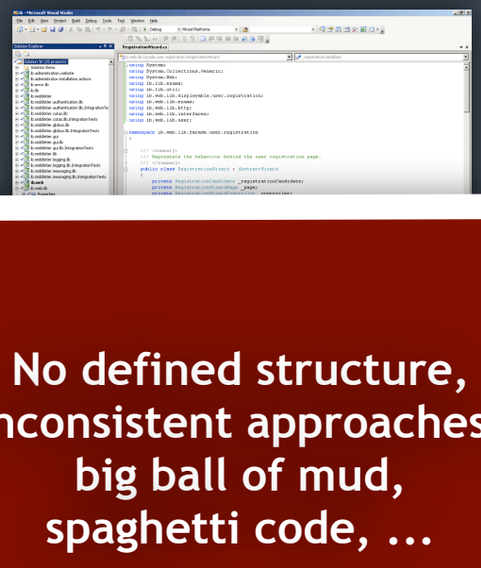


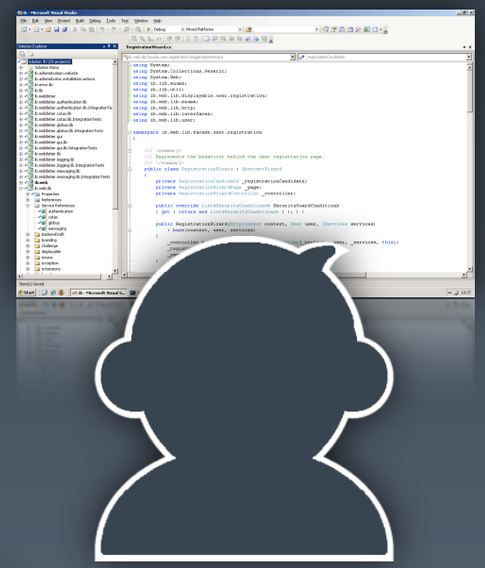
Chaos!

Does the team understand what they are building and how they are building it?





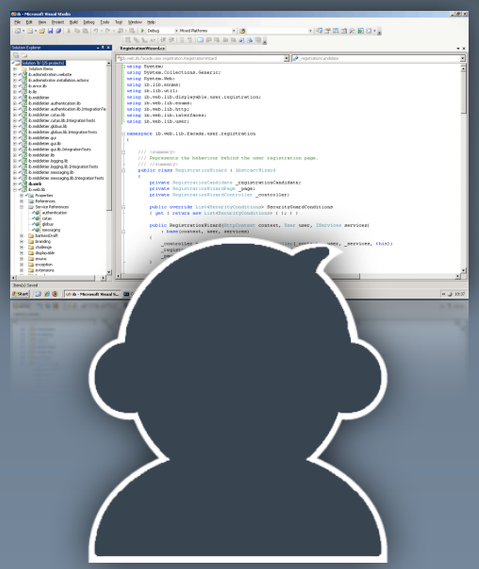


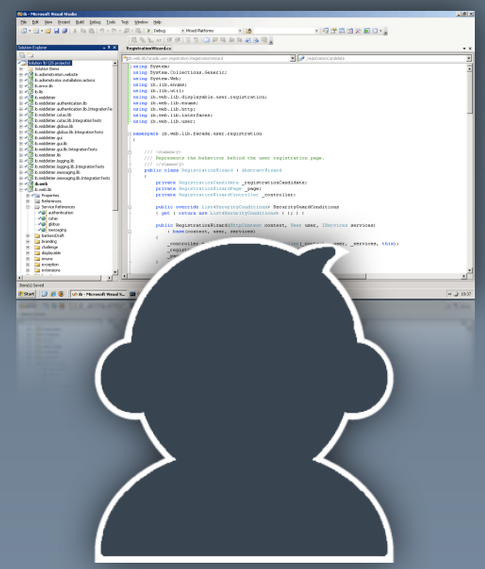


No defined structure,
inconsistent approaches,
big ball of mud,
spaghetti code, ...

STOP

Slow, insecure, unstable, unmaintainable,
hard to deploy, hard to change,
over time, over budget, ...





Sits in an ivory tower

Architect

Gap

Focusses on the low level detail

Developer

Developer

Developer

Developer

Developer

Steve Streeting
@stevestreeting

@simonbrown "Establish how to avoid ivory towers.": don't have a job title called 'Software Architect' ;)

Simon Brown
@simonbrown

@jose_casal @stevestreeting if all developers understood "software architecture", then we wouldn't need to differentiate, right? ;-)

Jose Casal
@jose_casal

@simonbrown @stevestreeting Absolutely. Wished we trained all Devs & Archs to master the Dark Arts of EDUF (and similar) :-)

Architect

Increased architectural awareness

Collaborating, coaching and mentoring

Reduced gap

Developer

In order to avoid the sort of struggle late Renaissance artists had to be recognised as more than mere artisans it would seem that master masons perpetuated a myth (as I see it) of being the descendants of noblemen. Further to this, by shrouding their knowledge with secrecy

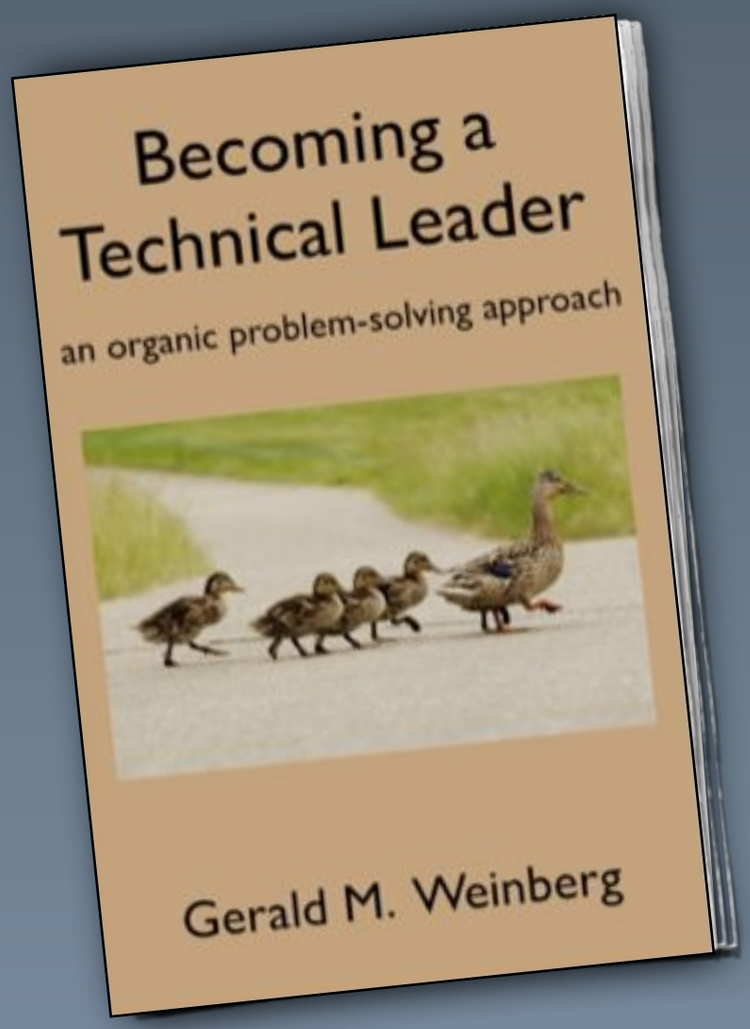
they created a mystique that separated them from other less 'arcane' or 'noble' professions.

jasongorman
@jasongorman

"I don't write code any more, that's how good I am at programming". We have some funny ideas about seniority in our industry

Would you hire a software architect that wouldn't code?

"Understanding the experiences of others expands our choices"

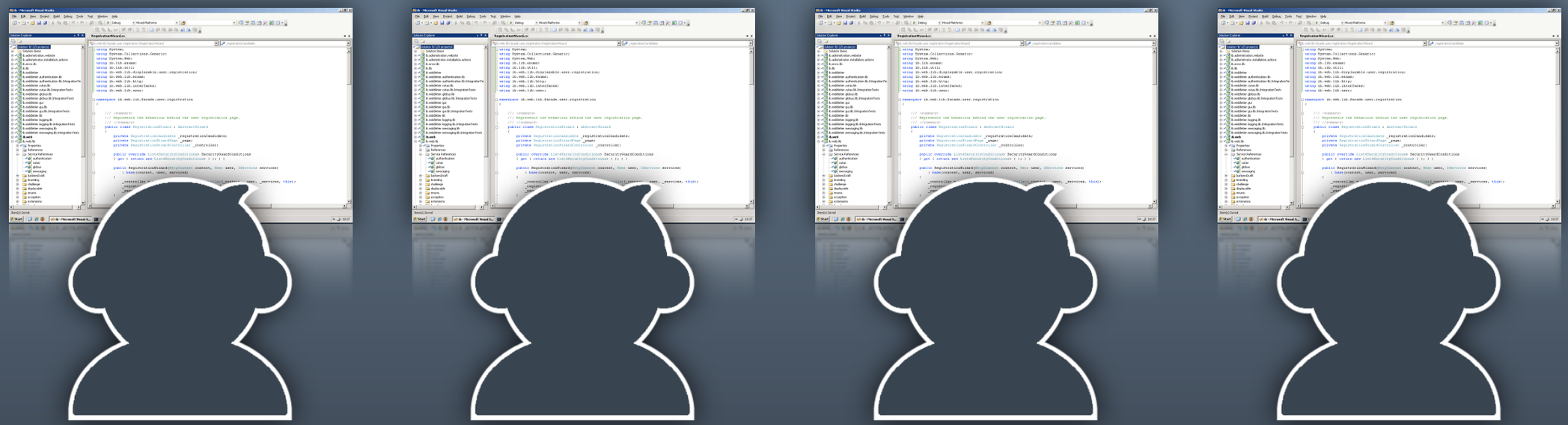


Becoming a Technical Leader

An organic problem-solving approach

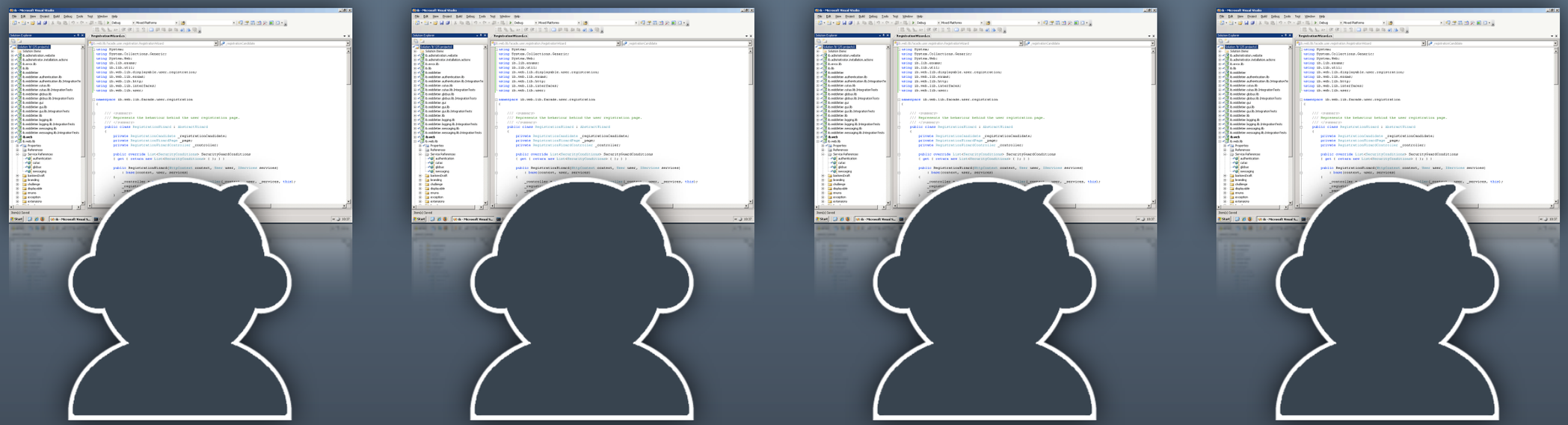
Gerald M. Weinberg

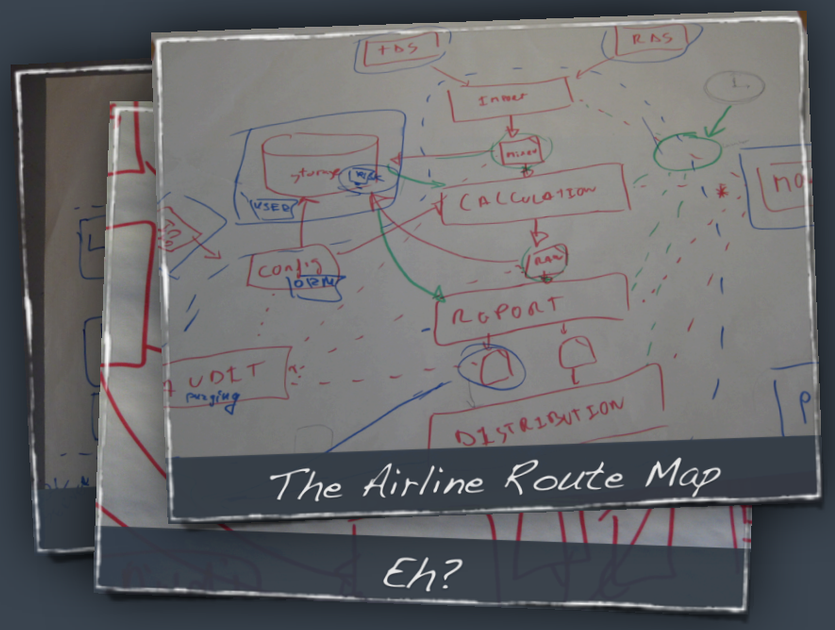
Shared vision



Shared vision of

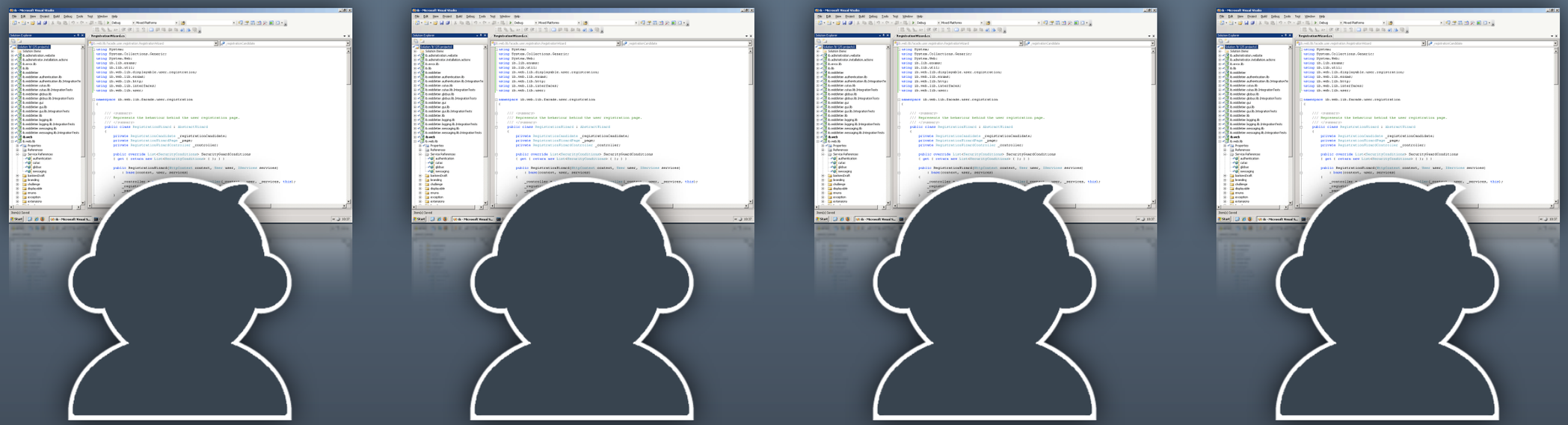
TL; DR



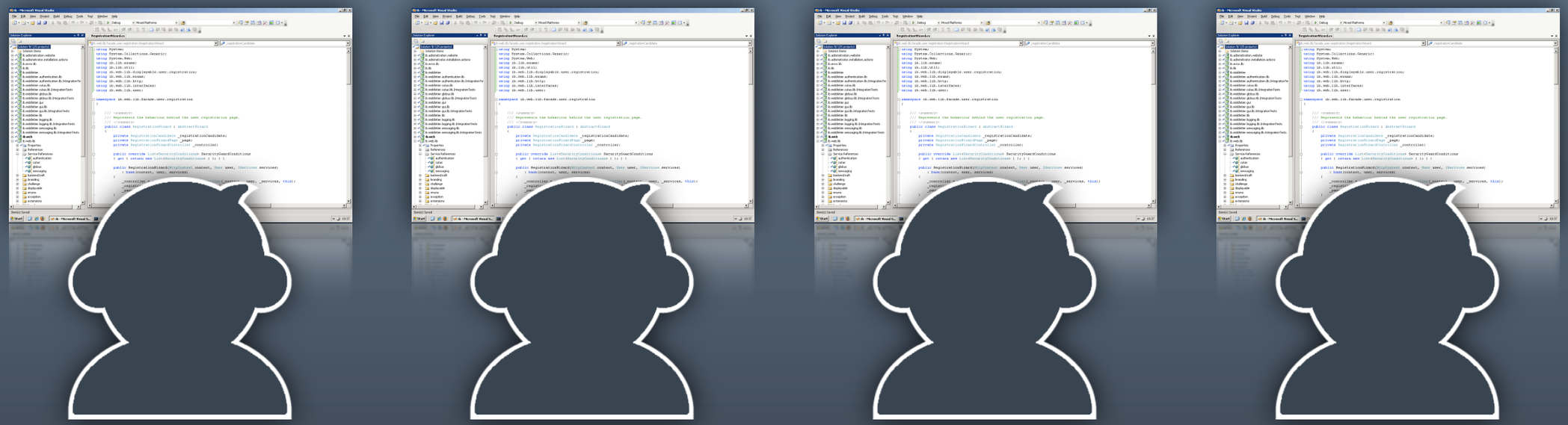


Shared vision of

WTF?!



Dandelions



Pictures provide an excellent way to create a

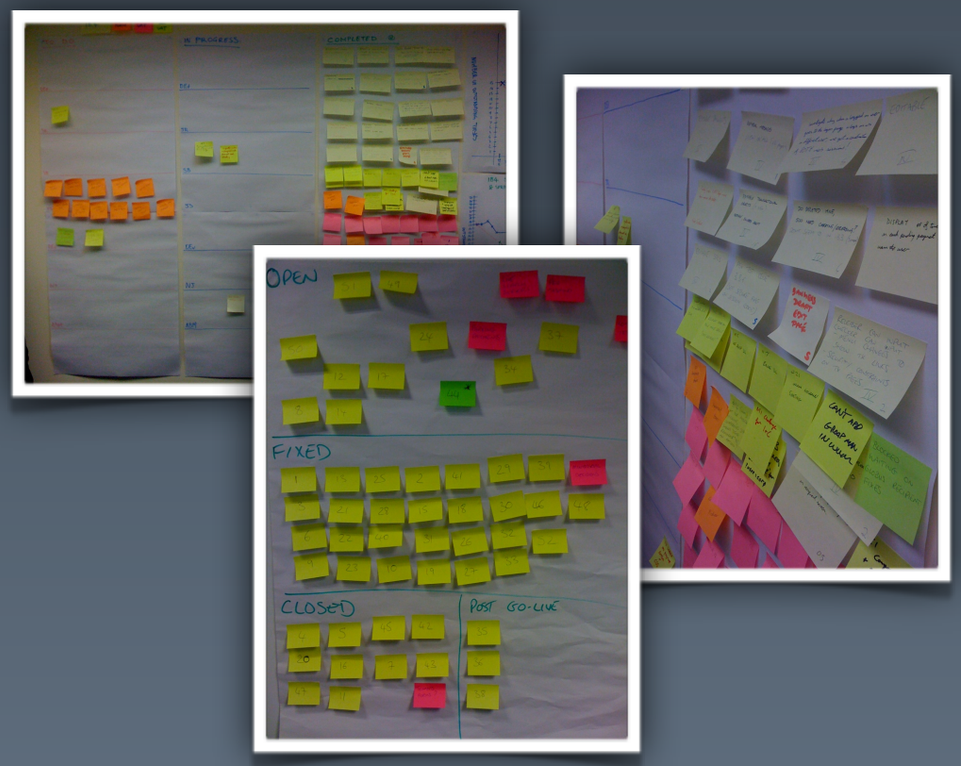
shared vision

for the team

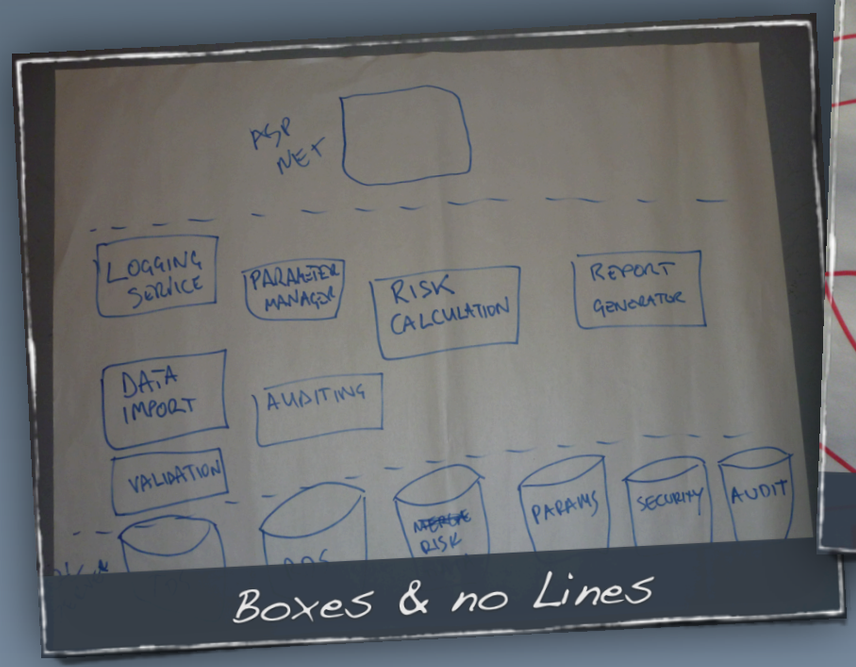
Making the implicit,

explicit

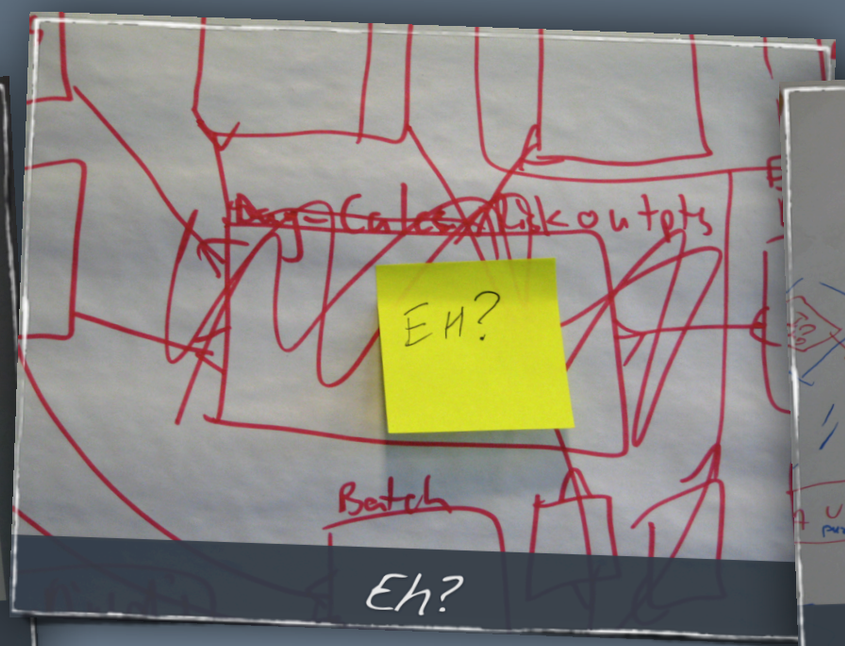
We can visualise our process...



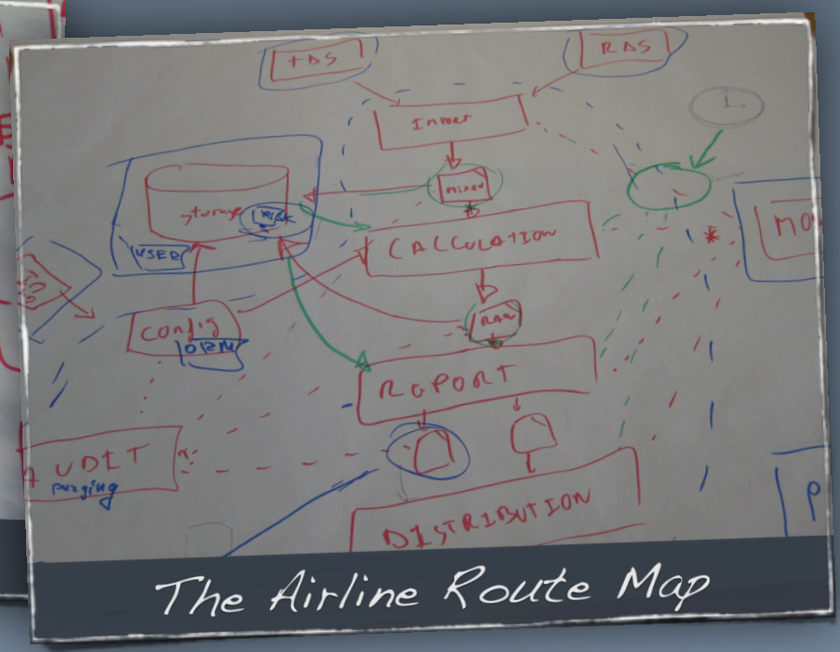
...but not our software!



Boxes & no Lines



Eh?



The Airline Route Map

Common Services | More Common Services | Service A | Service B | Service C

3rd Party Interface

Presentation Layer (Master Pages) | Presentation Layer (Service A Pages) | Presentation Layer (Service B Pages) | Presentation Layer (Service C Pages) — Presentation Layer

Service Proxy | Service Proxy | Service Proxy | Service Proxy | Service Proxy

WCF Communications (SOAP, REST, TCP, MQ) — Transport

Services Implementation | Services Implementation | Services Implementation | Services Implementation | Services Implementation — WCF Service

Contracts | Contracts | Contracts | Contracts | Contracts — Service Interface

Services | Services | Services | Services | Services | Services | Service A Services | Service B Services | Service C Services — Business Layer

Data Contracts | Data Contracts | Data Contracts | Data Contract | Data Contract | Data Contracts — Data Interface

Data Layer | Data Layer | Data Layer | Data Layer | Data Layer | Data Layer | Service A Data Layer | Data Layer | Data Layer | Service C Data Layer — Data Layer

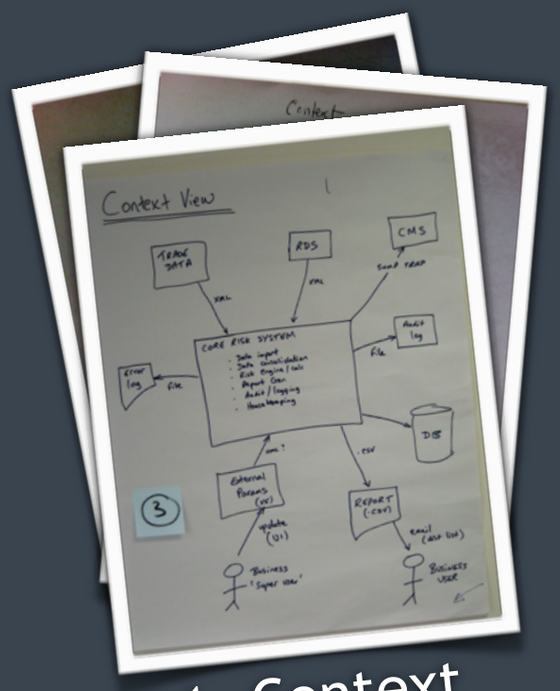
3rd Party Interface | 3rd Party Interface | 3rd Party Interface

Common

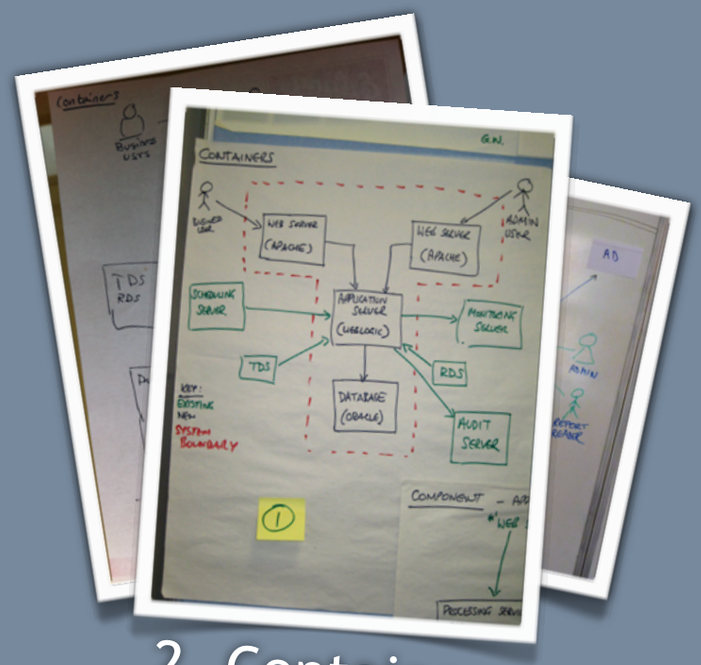
Authentication | Logging

Data Management | Data Access | CRM Workflow | Email

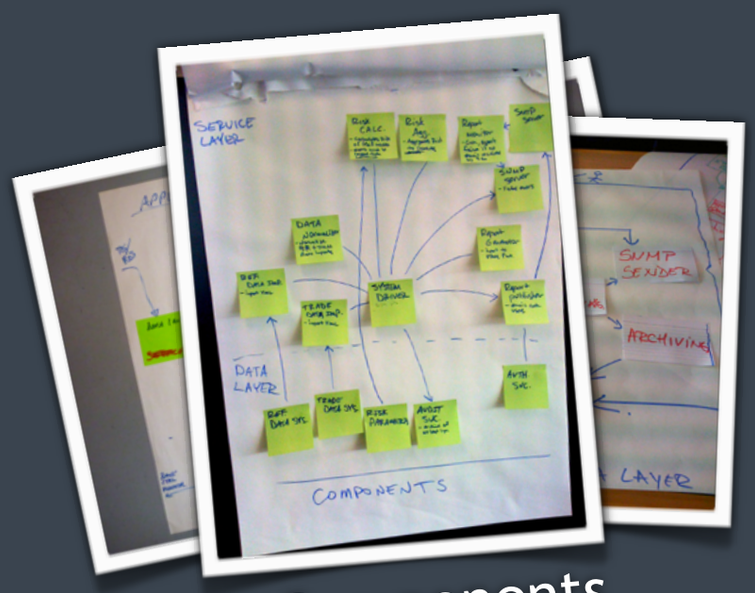
Foundation



1. Context



2. Containers



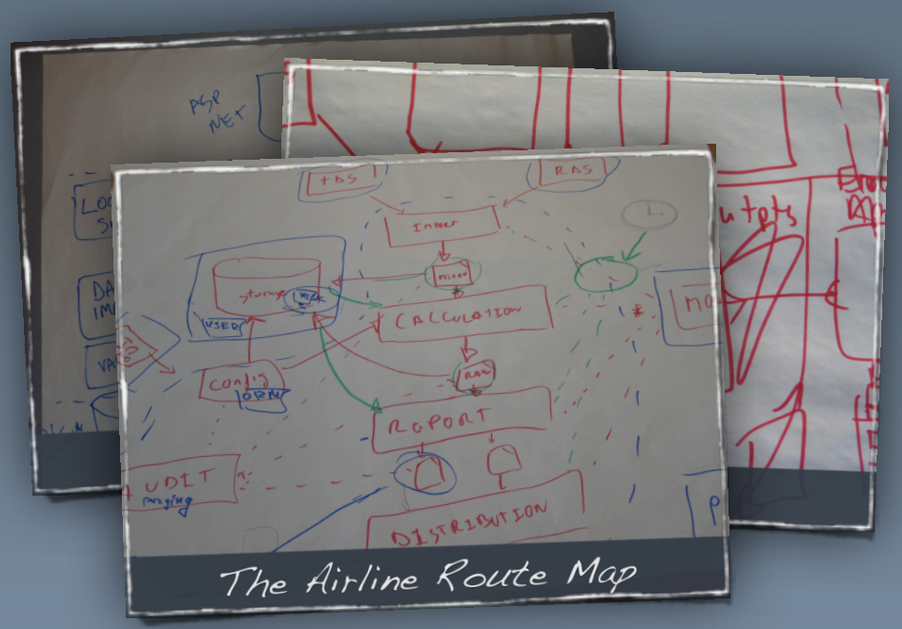
3. Components

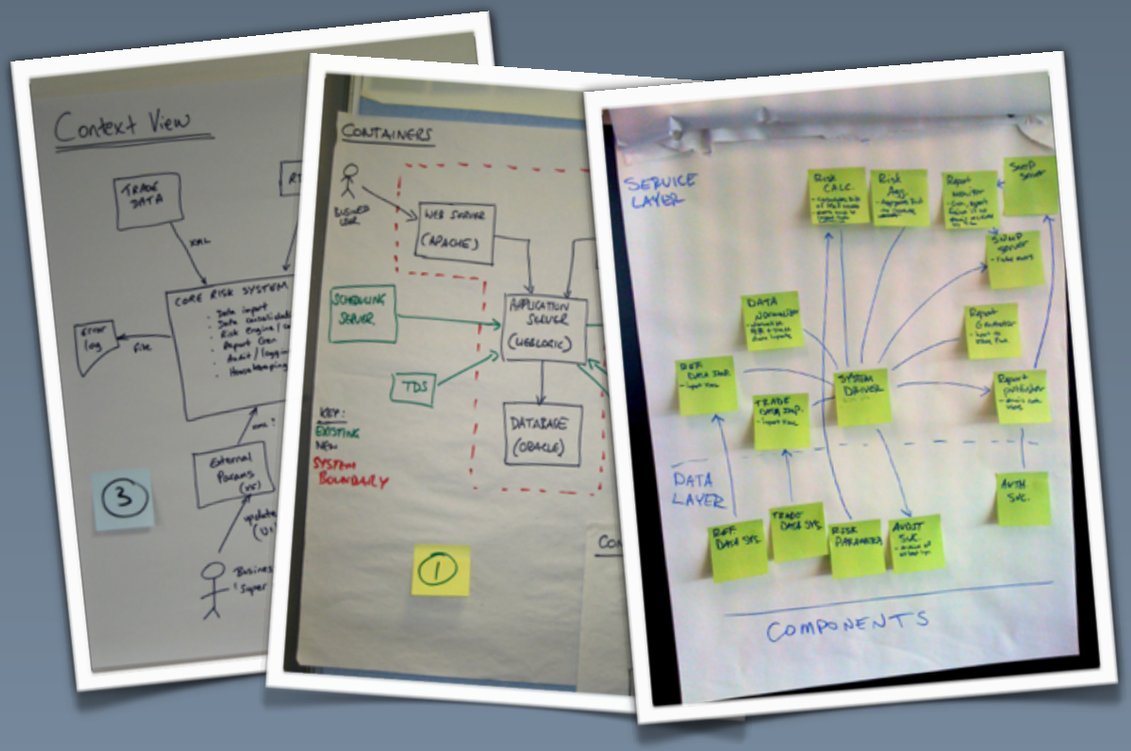
… and, optionally,
4. Classes

Thinking inside the box

Moving fast requires

good communication



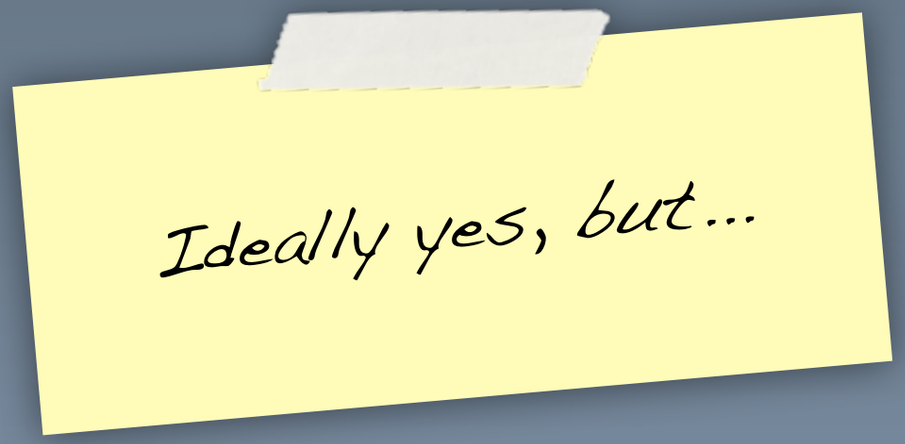


Should software architects write code?

Should software architects write

code

on software projects?



Generalising

Specialist

Depth

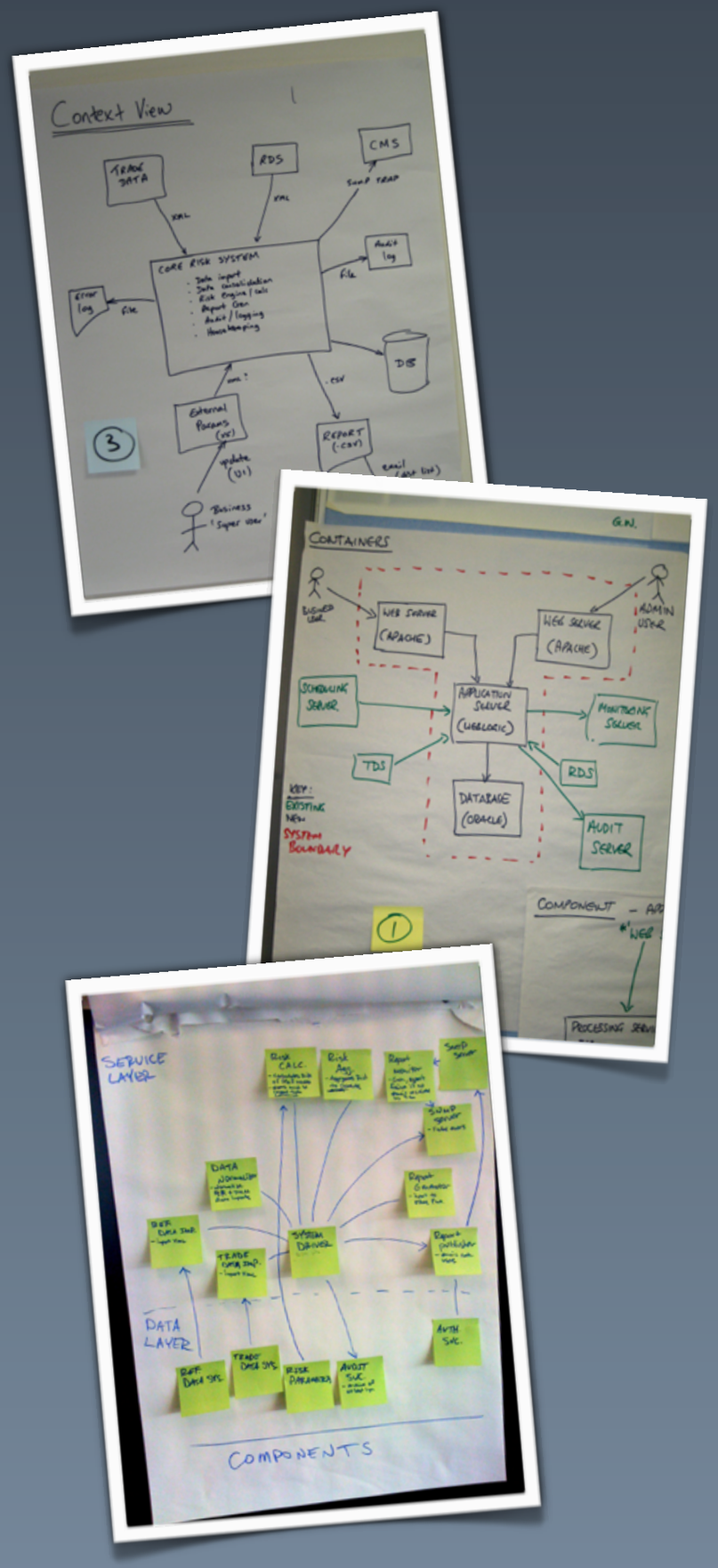
Deep hands-on technology skills and knowledge

Good software architects are master builders

Breadth

Broad knowledge of patterns, designs, approaches, technologies, ...

Awareness of options and trade-offs



Would we code it that way?

This is why

software architects must be able to code!

All decisions involve a trade-off

Big enterprise Java system

IBM WebSphere

An Enterprise JavaBean (EJB) wants to read files from the file system

File System

Option 1

Write a J2EE Connector Architecture (JCA) connector

Option 2

Use `java.io` from the Enterprise JavaBean

ORACLE Sun Developer Network (SDN)

APIs Downloads Products Support Training Participate

SDN Home > Products & Technologies > Java Technology > Reference > BluePrir

FAQ
EJB Restrictions

Are the restrictions mandatory? How are they enforced?

The restrictions are mandatory, since they are laid out in the specification as not o

- ***Compatibility Test Suite (CTS) and branding.*** Only servers that pass the CT branding control is to ensure the highest level of compatibility between vend implementations. The CTS and J2EE branding control indicate that the branc enterprise beans.
- ***Pre-deployment checking tools.*** Enterprise development tools have varying (meaning they abide by the restrictions). Developers can then be sure they are not deploying
etc.
- ***Security model restriction enforcement.*** The EJB specification defines that the EJB container implementation should enforce the restrictions described above by way of the standard Java security model. For example, an enterprise bean that tries to create an AWT `Frame` will fail, since the default security setting for the container denies `java.awt.AWTPermission`.

Specifically, enterprise beans should not:

- use the `java.lang.reflect` Java Reflection API to access information unavailable
- read or write nonfinal static fields
- use `this` to refer to the instance in a method parameter or result
- access packages (and classes) that are otherwise made unavailable by the rules of
- define a class in a package
- use the `java.awt` package to create a user interface
- create or modify class loaders and security managers
- redirect input, output, and error streams
- obtain security policy information for a code source
- access or modify the security configuration objects
- create or manage threads
- use thread synchronization primitives to synchronize access with other enterprise be
- stop the Java virtual machine
- load a native library
- listen on, accept connections on, or multicast from a network socket
- change socket factories in `java.net.Socket` or `java.net.ServerSocket`, or cha
- directly read or write a file descriptor
- create, modify, or delete files in the filesystem
- use the subclass and object substitution features of the Java serialization protocol

For more on these restrictions, see § 18.1.2 of the EJB 1.1 specification.

Why can't EJBs read and write files and directories in the filesystem? And why can't they access file descriptors?

Enterprise beans aren't allowed to access files primarily because files are not transactional resources. Allowing EJBs to access files or directories in the filesystem, or to use file descriptors, would compromise component distributability, and would be a security hazard.

Another reason is deployability. The EJB container can choose to place an enterprise bean in any JVM, on any machine in a cluster. Yet the contents of a filesystem are not part of a deployment, and are therefore outside of the EJB container's control. File systems, directories, files, and especially file descriptors tend to be machine-local resources. If an enterprise bean running in a JVM on a particular machine is using or holding an open file descriptor to a file in the filesystem, that enterprise bean cannot easily be moved from one JVM or machine to another, without losing its reference to the file.

Furthermore, giving EJBs access to the filesystem is a security hazard, since the enterprise bean could potentially read and broadcast the contents of sensitive files, or even upload and overwrite the JVM runtime binary for malicious purposes.

Files are not an appropriate mechanism for storing business data for use by components, because they tend to be unstructured, are not under the control of the server environment, and typically don't provide distributed transactional access or fine-grained locking. Business data is better managed using a persistence interface such as JDBC, whose implementations usually provide these benefits. Read-only data can, however, be stored in files in a deployment JAR, and accessed with the `getResource()` or `getResourceAsStream()` methods of `java.lang.Class`.

http://java.sun.com/blueprints/qanda/ejb_tier/restrictions.html

IBM

IBM WebSphere Application Server

Existing licenses, supports messaging, big, complex

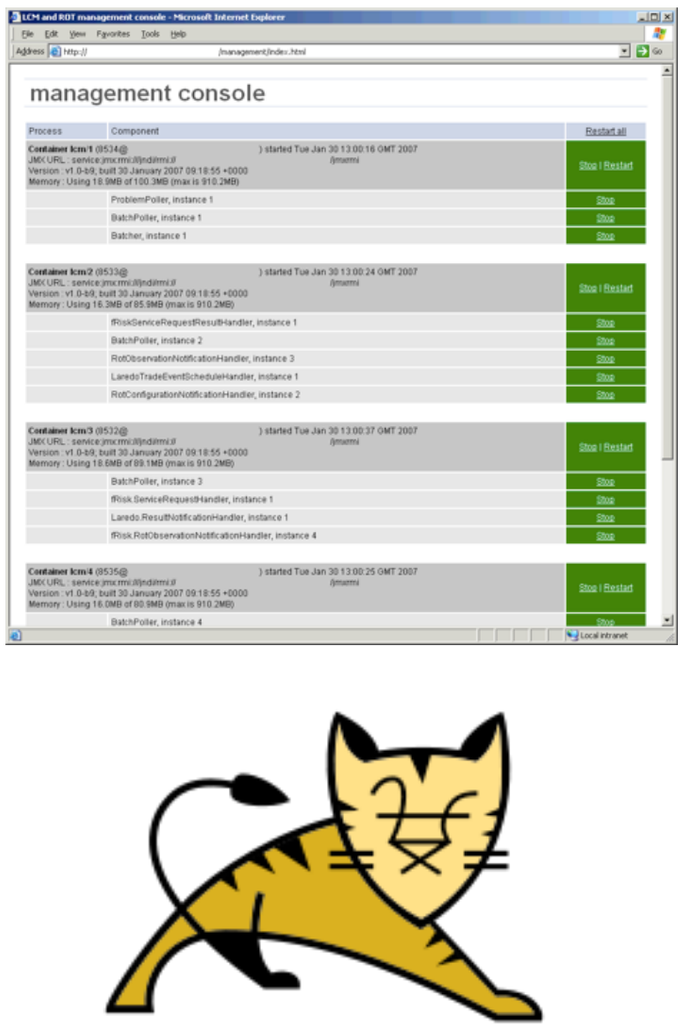
Apache Tomcat

Free, lightweight, not really designed for messaging

Standalone Java Application(s)

Lightweight, each JVM could be a message listener/processor

Trade-offs are deployment and operational complexity


LCM and ROT management console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http:// /management/index.html
management console
Process
Component
Restart all
Container lcm/1 (8534@) started Tue Jan 30 13:00:16 GMT 2007
JMX URL: service:jmx:rmi:///jndi/rmi:// /jmxrmi
Version : v1.0-b9; built 30 January 2007 09:18:55 +0000
Memory : Using 18.9MB of 100.3MB (max is 910.2MB)
Stop | Restart
ProblemPoller, instance 1
BatchPoller, instance 1
Batcher, instance 1
Container lcm/2 (8533@) started Tue Jan 30 13:00:24 GMT 2007
JMX URL: service:jmx:rmi:///jndi/rmi:// /jmxrmi
Version : v1.0-b9; built 30 January 2007 09:18:55 +0000
Memory : Using 16.3MB of 85.9MB (max is 910.2MB)
fRiskServiceRequestResultHandler, instance 1
BatchPoller, instance 2
RotObservationNotificationHandler, instance 3
LaredoTradeEventScheduleHandler, instance 1
RotConfigurationNotificationHandler, instance 2
Container lcm/3 (8532@) started Tue Jan 30 13:00:37 GMT 2007
JMX URL: service:jmx:rmi:///jndi/rmi:// /jmxrmi
Version : v1.0-b9; built 30 January 2007 09:18:55 +0000
Memory : Using 18.6MB of 89.1MB (max is 910.2MB)
BatchPoller, instance 3
fRisk.ServiceRequestHandler, instance 1
Laredo.ResultNotificationHandler, instance 1
fRisk.RotObservationNotificationHandler, instance 4
Container lcm/4 (8535@) started Tue Jan 30 13:00:25 GMT 2007
JMX URL: service:jmx:rmi:///jndi/rmi:// /jmxrmi
Version : v1.0-b9; built 30 January 2007 09:18:55 +0000
Memory : Using 16.0MB of 80.9MB (max is 910.2MB)
Stop | Restart
BatchPoller, instance 4
Stop
Local intranet


Java

JMX/RMI


Java


Java

Customer

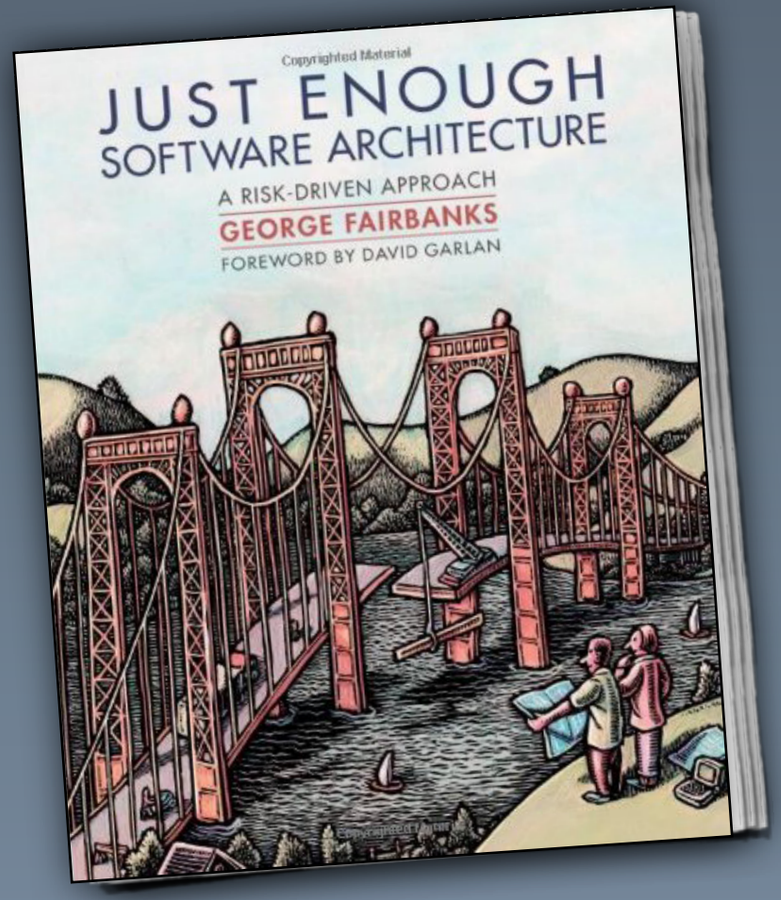
Software Architect

We need a system built

quickly

It’s possible if we don’t use the approved technology list

"architecture-indifferent design"



Just Enough Software Architecture

A Risk-Driven Approach

George Fairbanks

The Conclusion

Building architects were master builders,

therefore

software architects should code
rather than sitting in ivory towers

The End?

Divergence of the master builder role

Throughout ancient and medieval history, most architectural design and construction was carried out by artisans, such as stone masons and carpenters,

rising to the role of master builder.

Until modern times there was no clear distinction between the architect and engineer.

In Europe, the titles "architect" and "engineer" were primarily geographical variations referring to the same person, often used interchangeably.

Structural engineering has existed since humans first started to construct their own structures. It became a more defined and formalised profession with the

emergence of the architecture profession as distinct from the engineering profession

during the industrial revolution in the late 19th century.

Until then, the architect and the structural engineer were usually one and the same - the master builder.

Only with the development of specialised knowledge of structural theories that emerged during the 19th and early 20th centuries did the professional structural engineer come into existence.

Architect

Interacts with client, requirements, design, aesthetics, layout, etc

Structural Engineer

Makes sure the building doesn't fall over!


Generalising
Specialist

Generalising Specialist
+ Specialist(s)

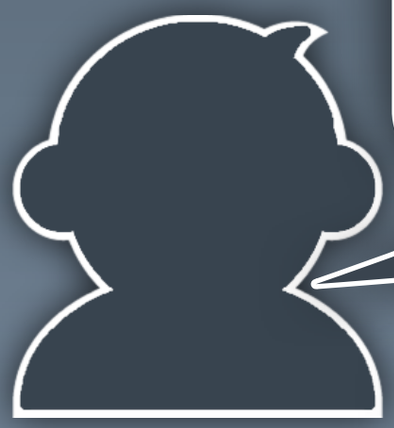
Generalist +
Specialist

Beware of
ivory towers!

Generalist +
Specialists

How can I *possibly* be
hands-on when I have

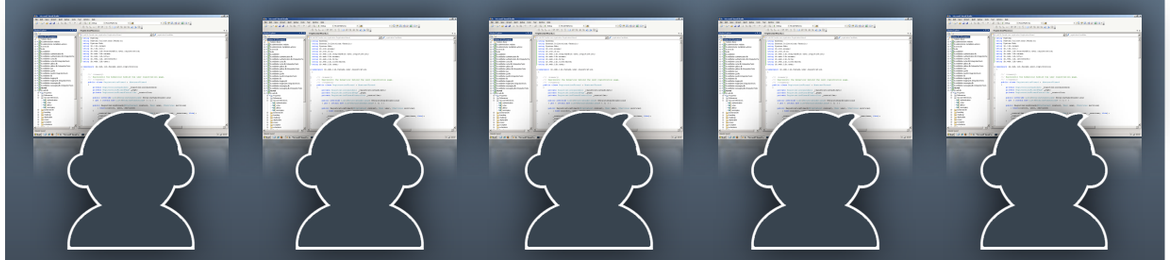
seven teams to look after?!

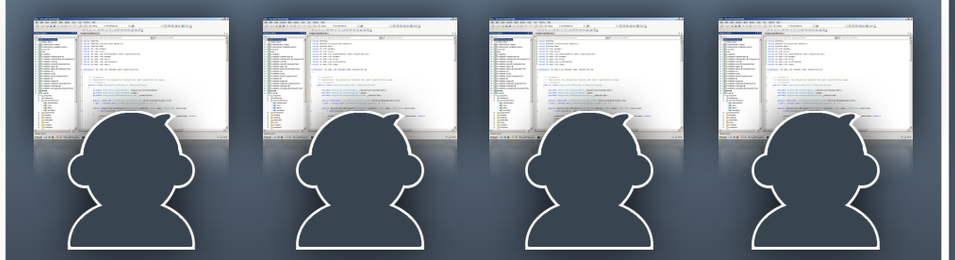


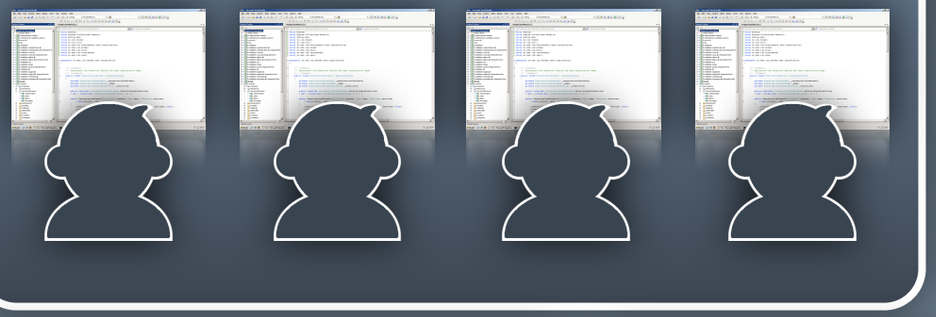
Software Architect

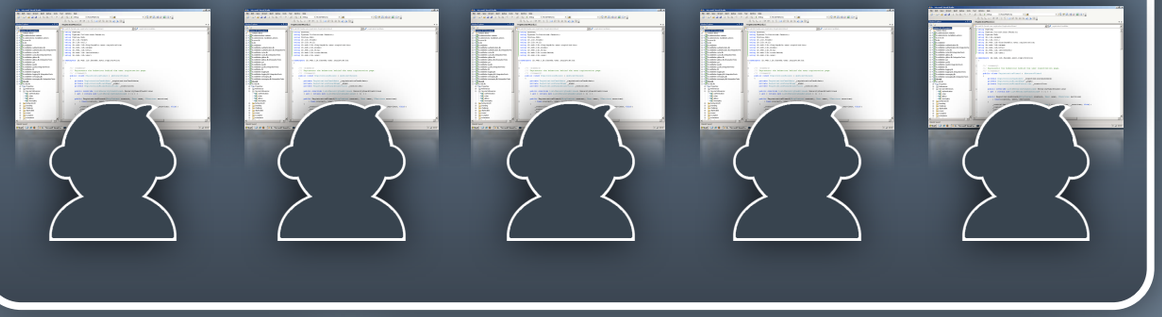
You can't ... but why are you looking after seven teams?

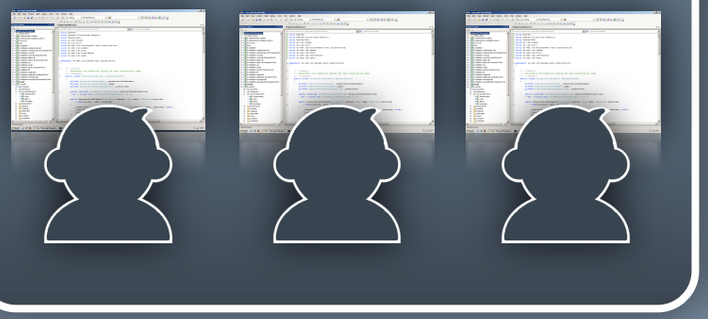
Enterprise Architecture Group

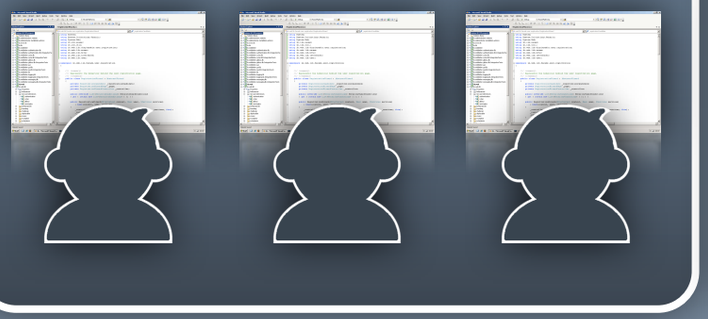


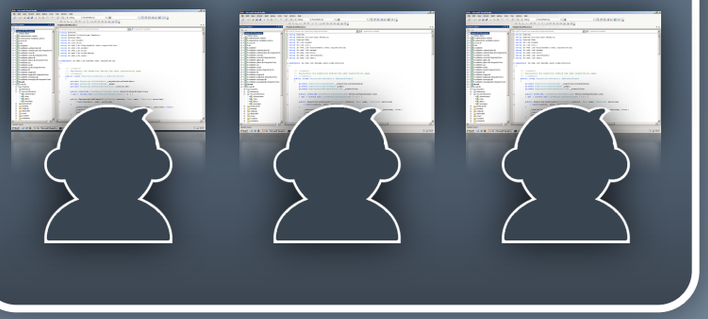






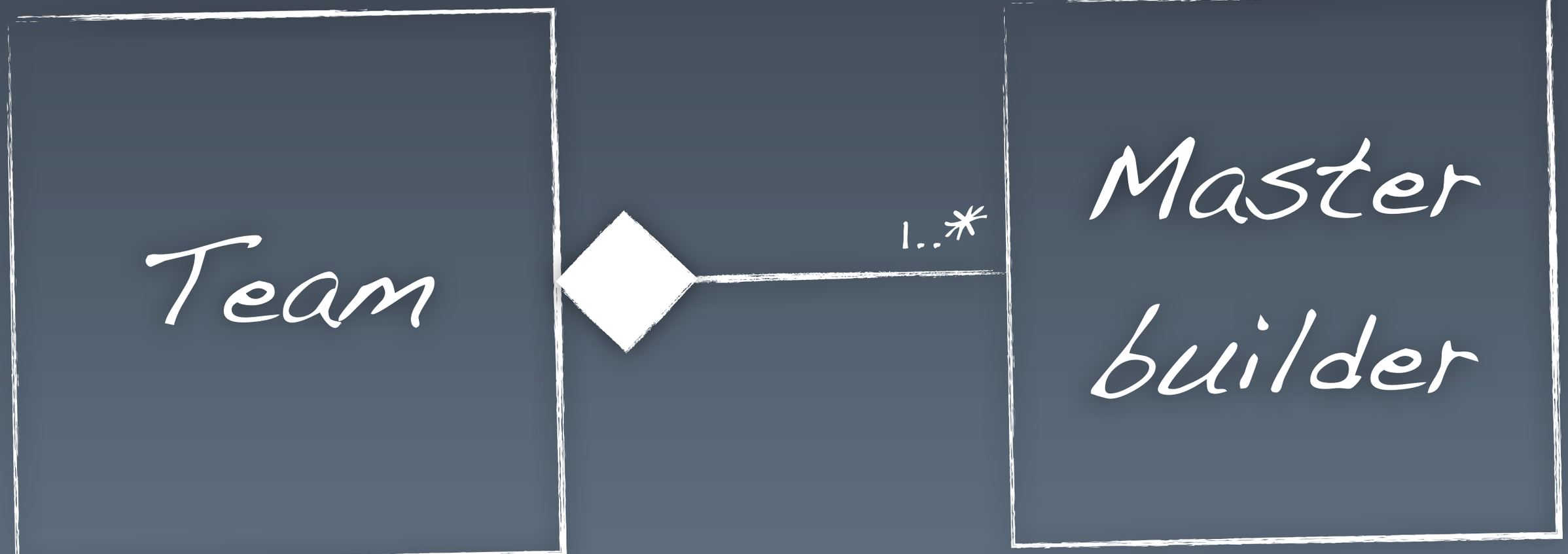






Although they may not have
touched stone, master masons

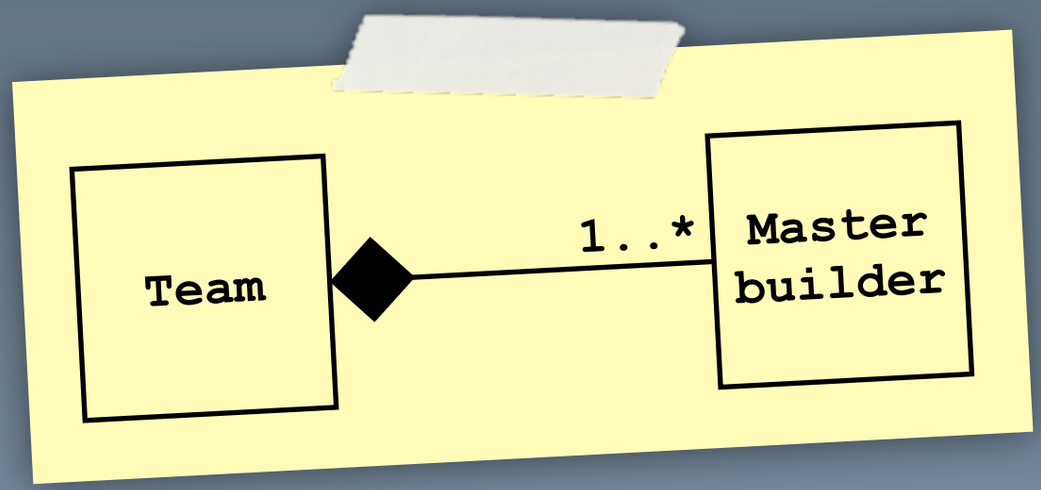
worked with the teams



This diagram might work, but many (most?) people in our industry don’t understand UML! :-/

Every software development team needs a master builder

1 or many



The software architecture role

From chaos to self-organising

Dedicated software architect

Single point of responsibility for the technical aspects of the software project

Elastic Leadership (Roy Osherove)
Chaos (command and control),
learning (coaching),
self-organising (facilitation)

Everybody is a software architect

Joint responsibility for the technical aspects of the software project

Corporate career ladders

Is coding a commodity?

The "corporate career ladder"

Non-technical

Technical

Medieval stonemasons' skills were in high demand, and members of the guild, gave rise to three classes of stonemasons:

apprentices, journeymen, and master masons.

Apprentices were indentured to their masters as the price for their training, journeymen had a higher level of skill and could go on journeys to assist their masters, and master masons were considered freemen who could travel as they wished to work on the projects of the patrons.

A mason would have an

apprentice

working for him.

Traditionally medieval stonemasons served a seven-year apprenticeship.

LinkedIn Account Type: Basic | Upgrade

Home Profile Contacts Groups Jobs Inbox 9 Companies News More Groups

97 Things Every Software Architect Should Know

97 Things Every Software Architect Should Know

Discussions Members Promotions Jobs Search More...

I am a senior developer. Recently, I was promoted to the position as architect. Could anyone please let me know which tools/software an architect should master/be familiar with. Thank you

10 months ago

Like Comment Follow Flag More

This is common :-/

We need to grow the

software architects

of tomorrow

To become a successful master mason in the medieval era, whatever one's origins or education, patently required a man of immense talent. Such a man needed to be a master in the

handling of physical and human resources

- to be able to direct them with imagination, ingenuity and efficiency toward a mercurial goal always framed by a host of compromises.

The opportunities for such a man whose creativity was matched by his organisational and diplomatic ability were varied and great.

Leadership and soft skills?

A 21st century software architect

The software architecture role

Architectural Drivers
Understanding requirements and constraints

Technology Selection
Choosing and evaluating technology

Architecting
Designing software

Architecture Evaluation
Understanding that the architecture works

Coding
Involvement in the hands-on elements of software delivery

Architecture Evolution
Ownership of the architecture throughout the delivery

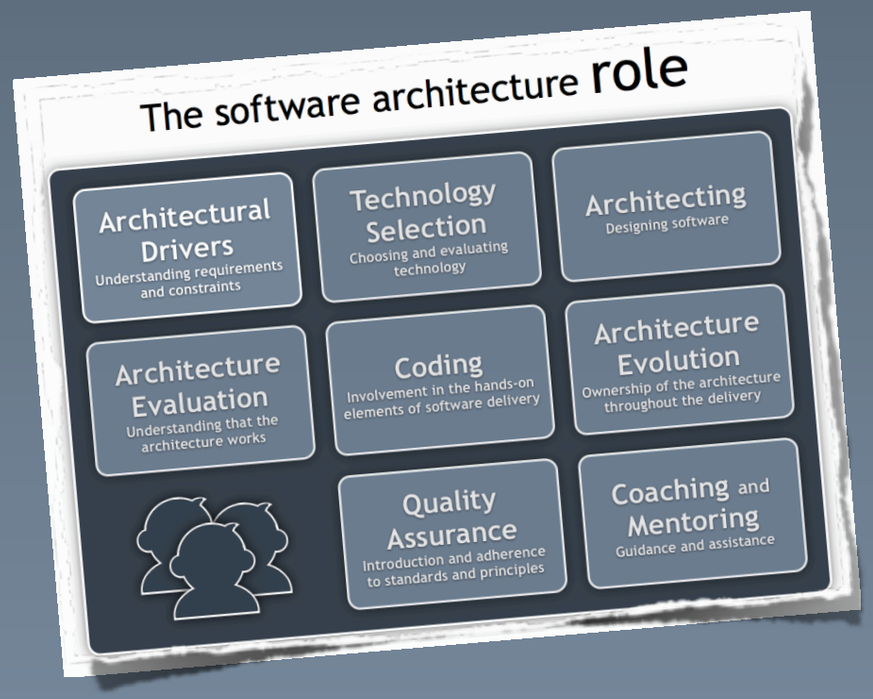
Quality Assurance
Introduction and adherence to standards and principles

Coaching and Mentoring
Guidance and assistance

Is a collaborative and lightweight approach to software architecture

the **missing piece**

of the **jigsaw**?



The process of software architecting

From big design up front to evolutionary

```
/// <summary>
/// Represents the behaviour behind the ...
/// </summary>
public class SomeWizard : AbstractWizard
{
    private DomainObject _object;
    private WizardPage _page;
    private WizardController _controller;

    public SomeWizard()
    {
    }

    ...

}
```

Big up front design

Requirements capture, analysis and design complete before coding starts

Evolutionary architecture

The architecture evolves secondary to the value created by early regular releases of working software

The role

“Just enough”

Understand how the significant elements fit together

Identify and mitigate the key risks

Provide firm foundations and a vision to move forward

The process

```
/// <summary>
/// Represents the behaviour behind the ...
/// </summary>
public class SomeWizard : AbstractWizard
{
    private DomainObject _object;
    private WizardPage _page;
    private WizardController _controller;

    public SomeWizard()
    {
    }

    ...

}
```

The Conclusion

Software development teams need

one or more

master builders

Number 1

The Conclusion

The master builders are in a leadership role that should be collaborative and include coaching

Number 2

The Conclusion

Effective and efficient communication

is a key skill for master builders

Number 3

codingthearchitecture.com

Training and consulting

leanpub.com Leanpub

simon.brown@codingthearchitecture.com
@simonbrown on Twitter

References

Analyse the role of the master mason in the C12-14th
http://www.moonshadow.co.uk/?p=66

Architect
http://en.wikipedia.org/wiki/Architect

BuildingArchitect
http://martinfowler.com/bliki/BuildingArchitect.html

Medieval Masons
http://www.historylearningsite.co.uk/medieval_masons.htm

Stonemasonry
http://en.wikipedia.org/wiki/Stonemasonry

Structural engineering
http://en.wikipedia.org/wiki/Structural_engineering

The Medieval Stonemason and the Master Mason
http://rachel-bellerby.suite101.com/the-medieval-stonemason-and-the-master-mason-a65816