Software Architecture for Developers

(five things every developer should know about software architecture)

@simonbrown

Simon Brown

Independent consultant specialising in software architecture, plus the creator of the C4 model and Structurizr

@simonbrown

1. Software architecture isn't about big design up front

Historically there’s been
a tendency towards
big design up front

“Waterfall”

Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

> I believe in this concept, but the implementation described above is risky and invites failure.

Managing the development of large software systems

Dr Winston W. Royce

Responding to change
over
following a plan

Big design up front

Software Architecture Document

vs

No design up front

“Big design up front is dumb. Doing no design up front is even dumber.”

Dave Thomas

I’m referring to **technical design** rather than product design

How much **up front design** should you do?

“it depends”

Sometimes requirements are known, and sometimes they aren’t

(enterprise software development vs product companies and startups)

“just enough”

Up front design is not necessarily about creating a perfect end-state or complete architecture

Evolutionary Design

Beginning With A Primitive Whole

A **starting point**
adds value

If you don’t **engage** in the problem, you end up with a very simplified and superficial view of the solution

1. Is that what we're going to **build**?

2. Is it going to **work**?

Architecture represents the **significant decisions**, where significance is measured by **cost of change**.

Grady Booch

Architecture

Programming languages
Technologies and platforms
Monolith, microservices or hybrid approach

Design

Implementation

Curly braces on the same or next line
Whitespace vs tabs

“Base your architecture on requirements, travel light and prove your architecture with concrete experiments.”

Agile Architecture: Strategies for Scaling Agile Development

Scott Ambler

Identify and **mitigate**
your **highest priority risks**

Risk-storming

A visual and collaborative technique for identifying risk

Concrete experiment

Proof of concept, prototype, spike, tracer, vertical slice, walking skeleton, executable reference architecture, ...

Enough up front design
to create a good
starting point and **direction**

Thinking about
software architecture lets you
stack the odds of success
in your favour

2. Every software team needs to consider software architecture

What happens if a software development team **doesn’t think about architecture**?

Chaos

Big ball of mud, spaghetti code, inconsistent approaches to solving the same problems, quality attributes are ignored, deployment problems, maintenance issues, etc

Every team needs

technical leadership

(irrespective of team size)

3. The software architecture role is about coding, coaching and collaboration

Software development is not a relay sport

AaaS

Architecture as a Service

Continuous technical leadership

Different types of teams need different leadership styles

Pair architecting

Soft skills

(leadership, communication, presentation, influencing, negotiation, collaboration, coaching and mentoring, motivation, facilitation, political, etc)

Talking with
Tech Leads
From Novices to Practitioners
Patrick Kua
Foreword by Jim Webber

Should software architects write **code**?

Production code, prototypes, frameworks, foundations, code reviews, experimenting, etc

Good software architects
are typically
good software developers

Software architects should be **master builders**

Progress Toward an Engineering Discipline of Software

Mary Shaw

The people designing software must understand technology …

all decisions involve trade-offs

1. Is that what we're going to **build**?

2. Is it going to **work**?

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

Experience is important ...

software architecture is not a rank!

The software architecture role is **multi-faceted**

(technical depth, technical breadth, soft skills)

4. You don't need to use UML

Do **you** use UML?

In my experience, optimistically,

1 out of 10 people use UML

UML, Cucumber and modeling reality - MPJ's Musings - Fun Fun Function

1:16 / 12:48

UML, Cucumber and modeling reality - MPJ's Musings - Fun Fun Function

1:42 / 12:48

UNIX BOX
RS
TRANSPORT + LOGIC
JBOSS INSTANCE
ERROR
JBOSS INSTANCE (WEB CONTAINER ONLY)
WINDOWS BOX
MS
SQL
SERVER
MS
REPORTING SERVICE

ASP NET
LOGGING SERVICE
PARAMETER MANAGER
RISK CALCULATION
REPORT GENERATOR
DATA IMPORT
AUDITING
VALIDATION
TDS
RDS
~~MERGE~~ RISK DATA
PARAMS
SECURITY
AUDIT

FUNCTIONAL VIEW

File Retriever

Scheduler

Auditing

Reference Archiver

Risk Assesment Processor

Risk Parameter Configuration

Report

Import
mixer
storage
RISK
USER
CALCULATION
config
ORM
RAW
REPORT
AUDIT
purging

TDS
RDS
trade report
ref repo
RG
TRANSPORT
ARCHIVE
AUDIT
action
data
error
error
BUS. LOGIC
action
error
ERROR
MONITOR
REPORT GEN
action
error
Scheduler
config

2
SERVICE/BATCH
IMPORT
JOIN
CALC
EXPORT/STORE
REPORT
SNMP
CENTRAL
MONITORING
SERVICE
3
WEBAPP
•GET REPORT
•EDIT
RISK RULES
•AUTH
•AUDIT
RISK RULES

VIN SERVER
APPLICATION
CONSOLE - C#
SCHEDULER
CONSOLE - C#
DATA LOADER
C#
CALCULATION MANAGER
C#
NOTIFICATION MANAGER
C#
EXTERNAL SYSTEMS
SNMP SERVICE
ROS
IDS
check if alive
DATA MERGE C#
XPath
XML READER
C#
XML
RISK CALCULATION
C#
NOTIFICATION SERVICE
C#
SMTP
parameters
output of risk
LOGGER
.log
SNMP TRAP GENERATOR
DATA ACCESS
SQL SERVER

DATA POLL SERVICE
TRADES
REF DATA.
AUTO RE-START.
ARCHIVE
DATA READER TRADES
DATA VALIDATOR
reject
DATA READER REF DATA
DATA VALIDATOR
reject
MONITOR SERVICE SMTP TRAP
FAIL EVENTS
DATA STORE
DATA TRANSFORMATION INTEGRITY VALIDATION
rejects
tolerence.
FAIL EVENT
COMPLETE EVENT
STOP
STOPPED EVENT
COMPLETION EVENT
CALCULATION ENGINE
COUNT
A + B = C
reject.
FAIL EVENT
COMPLETE EVENT
OUTPUT STORE
FAILURE
PARAMS TOLER.
UI
SECURITY
AUDIT

ENGINE
APP SERVER
RING
WEB SERVER
REPORTING SERVICE

Params
Calcs
EH?
Floor
App Log
Batch
Date
- calcs
- Par
Params_Distrib
B_id:

Video over IP
Capturing video frames from Camera
Format conversion and Pre-processing
H.264 Encoder
NALU Fragmentation
Audio/Video sync module
Format conversion and Post-processing
H.264 Decoder
Reconstruction of NALU
camera
Display
RTP/ RTCP
Network Layers
TCP
UDP
IP
MAC
Ethernet
Voice over IP
Acoustic echo cancellation (AEC)
LEC (G.168/ G.165)
AGC - Gain Control
VAD
Speech Encoder
AGC - Gain Control
Speech Decoder / CNG
PLC
AJB
mic
Speaker
modem
Audio Interface layer
Sound card Interface (PCI / USB)
FXS/TDM Interface
Interface layer - tion, CPT/DTMF, MF-R1/R2, odem Discrimination Unit
Machine
Signing
Authentication
Authorization
Rich UI
Web UI
Controls
Service Interface
Activity
Business Rules
Human Workflow
Workflow
Service Agents
DAL
E-Publish
EAI
ECM
DW
OLTP
Monitoring
Log & Trace
Exception Management
Configuration
Session Initiation Protocol (SIP)
V²oIP
Video over IP
Voice over IP
Fax over IP
CA Server Automation
Administration UI
Reservation Manager UI
AutoShell
Packaging
Process Automation
Configuration Management / Compliance
Help Desk
Reporting (Jasper)
Agent Remote Deployment
Service Response Monitoring
CA IT Client Manager
CA Patch Manager
CA Process Automation
CA Configuration Automation
CA Service Desk
Chargeback
Agent Policy Configuration
Server Monitoring (Local & Remote)
Web Services and Reliable Transport (Active MQ)
Storage
Platform Management
Microsoft HyperV
Provisioning
Automation Infrastructure Platform
AOM Data Service
Policy, Performance, and Resource Management
Authentication / Security (CA EEM)
State and Event Management
Scheduler (Windows)
Discovery (CA Network Discovery Gateway)
Management DB
Performance DB
Racemi Dynacenter
Rapid Server Imaging
ASCII File Editor
ASCII Files
Graphics Editor
Library
SCADA Developt. Environ
Export / Import
Project Editor
Driver Toolkit
SCADA Client
HMI
Trending
Alarm Display
Log Display
Active X Controls
3rd Party Applic.
Active X Container
Client - Server - Publish / Subscribe - TCP/IP
SCADA Server
Recipe DB
Recipe Managt
Ref. DB
Data Process-ing
RT & Event Manager
RT DB
Report Gener.
Alarm
Log
Archive
SQL
Alarm DB
Log DB
Archive DB
ODBC
DDE
API/DLL
Data R/W
Driver
OPC
Private Application
EXCEL
VME
PLC
PLC
Typical generic software architecture of SCADA systems
Management Service
Management Service
Computational Analysis Service
e-AIRSview
CFD Solver
Output Converter
APSE
(Adaptive Problem Solving Environm
Remote Experiment Service
Exp. Exec.
Exp. Info.
Aerodynamic Exp.
PIV Exp.
Parametric Study
Validation
Analysis
Collaboration
Session Reservation
Session Management
Resource Configuration Service
Resource Monitoring Service
Authentication Service
Remote administration
Control interface
Input devices
Input device manager
Object-Tracker
Real-time kernel
User manager
DB-Server
Analysis
Logic Sublayer
XML / SOAP / HTTP
e-AIRS Middleware Service Layer
SASP (Scientific Application Service Provider)
Basic Execution Service
Parameter Sweeper
Global Scheduler
Resource Catalog
GRAM adapter
Plotting Service
eAIRS Plot
Conv. Graph
Data Transfer Service
Metadata Management Service
AG + VNC
Visualization Sharing
Mobile Service Component
A/V Conferen
Chat

If software developers created building architecture diagrams...

1. Is that what we're going to **build**?

2. Is it going to **work**?

Teams need a **ubiquitous language** to communicate effectively

The primary use for diagrams and documentation is **communication** and **learning**

Software architecture diagrams, when connected to the code, are also a fantastic tool for **architectural improvement**

A **common set of abstractions**
is more important
than a common notation

A **software system** is made up of one or more **containers**, each of which contains one or more **components**, which in turn are implemented by one or more **code elements**.

C4

Context, Containers, Components and Code

Diagrams are maps

that help software developers navigate a large and/or complex codebase

Diagrams are maps

that help software developers navigate a large and/or complex codebase

What **tools** do you recommend?

The **1990**’s called and they want their tools back!

It’s 2018 and we shouldn’t be using a general purpose diagramming tool for software architecture

c4model.com

(for more information about software architecture diagrams)

5. A good software architecture enables agility

Agile is about moving fast, embracing change, releasing often, getting feedback, ...

Agile is about a mindset of
continuous improvement

Agility is a

quality attribute

A good architecture enables agility

A good architecture rarely happens through **architecture-indifferent design**

Modular monolith
Microservices
Monolithic big ball of mud
Distributed big ball of mud
Modularity
Number of deployment units

Modular monoliths

Simon Brown
@simonbrown

I'll keep saying this ... if people can't build monoliths properly, microservices won't help. #qconlondon #DesignThinking #Modularity

Retweets **258**

Likes **109**

10:49 am - 4 Mar 2015

Architect Clippy
@architectclippy

I see you have a poorly structured monolith. Would you like me to convert it into a poorly structured set of microservices?

RETWEETS
4,441

LIKES
2,743

12:59 AM - 24 Feb 2015

Five things every developer should know about software architecture

1. Software architecture isn't about big design up front
2. Every software team needs to consider software architecture
3. The software architecture role is about coding, coaching and collaboration
4. You don't need to use UML
5. A good software architecture enables agility

simon.brown@codingthearchitecture.com
@simonbrown