Lean and Mean Architecting with RCDA

Eltjo R. Poort
SATURN 2013, Minneapolis
Eltjo Poort

CGI NL Lead Expert Architecture
- Reviewing Bids & Projects
- Standardising & Improving Architecture Practice in CGI NL

GGI Architecture Community of Practice lead

Researcher
- Improving Architecture Practices
- With Universities (VU, Twente, Utrecht, Eindhoven)
- Member if IFIP WG 2.10 Software Architecture
Back to Basics in Software Methodologies

…a process…that include[s] a kernel of widely-agreed elements, extensible for specific uses.

[SEMAT initiative, 2010]

Maybe we should try “lean and mean” software process models, rather than making them “richer.”….start from what people do … and not from what we think a priori they should be doing.

[Kruchten, 2011]

These calls resonate with the way Logica/CGI has been improving Solution Architecting practices since 2007.
Why *Solution* Architecture?

For some of the things we architect, “Software” Architecture is a stretch:

- software application
- software as a service
- embedded system
- system of systems
- systems integration
- BPO solution
- full service solution
- full IT outsourcing
- …
Solution Architecture approach

Key characteristic

A good Solution Architecture approach fills the gap between:

- Enterprise Architecture approaches
  - TOGAF, IAF, Zachman...
  - Strong on principles & governance
  - Weak on transformation & implementation

- Technical Architecture approaches
  - E.g. Software / Infra / SOA (RUP, ATAM…)
  - Strong on design & implementation
  - Weak on cross-technology stakeholder concerns
Risk- and Cost Driven Architecture
RCDA history until now

- **'07**
  - Risk- and Cost Driven Architecture initiative (Group Technical)
  - Process → set of practices

- **'09**
  - RCDA 1.0 released
  - Solution Architecture Practitioner Course (NL, >100 trained)

- **'10**
  - RCDA 1.1
  - Solution Architecture international training starts based on RCDA
  - Group-wide ratification

- **'11**
  - RCDA 1.2
  - Major presentation improvements
  - Key practice extension

- **'12**
  - RCDA 1.3
  - RCDA as CGI proposition

- **'13**
Basis of Risk- and Cost Driven Architecture

Extensive experience in Solution Assurance
• Reviewing 100s architectures in bids and projects

Industry Leading Architecture visions
• SEI, TOGAF, IBM

Research in Architecture Knowledge Management
• Focus on Architectural Concerns & Decisions

Feedback from CGI Architect Community Network
RCDA is a Practice repository

**Practice**: a light-weight, proven way of addressing a problem
- see e.g. EssUP (Ivar Jacobson)
- more pragmatic than processes
  - easier to maintain
  - more fine-grained
  - more flexible
- easily applicable in any process/organisation
  - fit in existing sales/design/development processes
  - avoid need to adopt complete new process

**Practice repository**: stores practice descriptions and associated documents (templates, guidelines, checklists, examples…)
Key Principles of RCDA

Cost and Risks drive architecture
- highest impact on *cost and risks* of the system and its delivery
- architect should be an expert on costing and risk mitigation

Architecture should be minimal
- to keep overview of the whole system
- Solution Architect should limit to decisions with critical impact
- leave a maximum of design space for developers

Architecture as both Blueprint and Decisions
- decisions leading to architecture and the underlying rationale are essential

Solution Architect as Decision Maker
- critical architectural decisions are made by *one person* with overview of whole system
- requires authority and subject matter skills and knowledge
What is architecture about?

“Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution”.

“Architecture is about the important stuff. Whatever that is.”

After talking to architects and stakeholders on dozens of projects, we have come to equate the “important stuff” with the stuff that has most impact on risk and costs.

Important ⟷ high risk and cost ⟷ architectural significance
The Architect’s Daily Job

Architecting Microcycle

- Identify & prioritise architectural concerns
- Research possible solutions
- Decide best fitting solution
- What problems should I work on?
- What are my options?
- I’ll pick this one
The Architecting Microcycle

- Identify & prioritise architectural concerns
- Research possible solutions
- Decide best fitting solution
The Architecting Workflow

- Identify & prioritise architectural concerns
- Research possible solutions
- Decide best fitting solution
- Architectural decisions
- Architectural concerns (backlog)
Building Up the Core Architecting Process

Core Practices

Architectural Requirements Prioritisation
Solution Selection
Applying Architectural Strategies
Architecture Documentation
Solution Costing
Architecture Evaluation
Architecture Implementation
Architecture Maintenance

- What problems should I work on?
- What are my options?
- I’ll pick this one
- Write down what I want
- What is this going to cost?
- Is this really going to work?
- Let’s do it!

Solution Architect
Adding Supporting Practices

Core Practices
- Architectural Requirements Prioritisation
- Solution Selection
- Applying Architectural Strategies
- Architecture Documentation
- Solution Costing
- Architecture Evaluation
- Architecture Implementation
- Architecture Maintenance

Supporting Practices
- Stakeholder Workshop
- Dealing with NFRs
- Requirements Convergence Plan
- Architecture Roadmapping
- Solution Shaping Workshop
- Cost-Benefit Analysis
- Documenting Architectural Decisions
- Independent Architecture Assessment
- Architectural Prototyping
- Supplier Evaluation
- Technical Debt Control
RCDA Practice Sets and Lifecycles

Core Practices
- Requirements Analysis
  - Architectural Requirements Prioritisation
- Solution Shaping
  - Solution Selection
  - Applying Architectural Strategies
  - Architecture Documentation
  - Solution Costing
- Architecture Validation
  - Architecture Evaluation
- Architecture Fulfillment
  - Architecture Implementation
  - Architecture Maintenance

Supporting Practices
- Stakeholder Workshop
- Dealing with NFRs
- Requirements Convergence Plan
- Architecture Roadmapping
- Solution Shaping Workshop
- Cost-Benefit Analysis
- Documenting Architectural Decisions
- Independent Architecture Assessment
- Architectural Prototyping
- Supplier Evaluation
- Technical Debt Control

Lifecycles
- RCDA Core Process
- Waterfall Project
- RUP Software Development
- Agile Development
- Bid
- Blended Delivery
- Enterprise to Solution
RCDA in Context

RCDA Solution Architecting

- Requirements Analysis
- Solution Shaping
- Architecture Validation
- Architecture Fulfillment

Stakeholders

- Higher Level Architecture

Product
- Roadmapping
- Requirements Definition
- Costing
- Solution Presentation

Project
- Risk Management
- Decision Management
- Requirements Management
- Realisation
- Transition
- Transformation

Service
- Solution Assurance

Environment

- Architectural Asset Base
- Architectural Asset Management

Actor
- Practice Set
- External Activity

Key:
- Responsible
- Primary flow

- Solution Architect
- Solution Reviewer
- Asset Architect
RCDA Survey Results
October 2011

- 60% of architects report increased effectiveness after RCDA training
- 77% of lead architects report increased effectiveness

RCDA is Lean, Mean and Agile

Practices (core and extensions) make RCDA Lean
Risk and Cost focus make RCDA Mean
Architectural Decision Microcycle makes RCDA Agile
Solution example: Mobile Application

Athletes use app to register their location for Anti-Doping Authority. Used by 58% of Dutch athletes at 2012 Olympics.

- App on Android, Apple, Blackberry.
- Server platform & back-office software.
- Maintenance & support.
Solution example: Software Application ($1.5M)
State Government Election Count System

Delivery Scope:
• Architect & design software & hardware platform
• Development (Microsoft .net)
• Installation and configuration on client’s hardware
• Warranty & support
Solution example: Outsourcing (70M€, 7yr)
Bugle Bank Full IT Outsourcing
Architecture Lifecycle
UP Software Development

**Inception/Elaboration phase**
- Dealing with NFRs
- Supplementary Specification
- Requirements Convergence Plan
- Elaboration Phase planning
- Architecture Documentation
- Software Architecture Document
- Architectural Requirements Prioritisation
- Solution Selection
- Applying Architectural Strategies
- Elaboration Phase exit criteria

**Construction/Transition phase**
- Architecture Roadmapping
- Iteration planning
- Architecture Implementation
- Ensuring compliance with SAD by developers
- Coordinating integration
- Testing architectural requirements
- Architecture Maintenance
- Managing architectural change
- Technical Debt Control
Architecture Lifecycle
Enterprise Architecture to Solution Architecture

• Solution: part of transformation under EA
  • Gap from actual to target enterprise architecture

• EA → Requirements + Architectural Guidance
  • Part of Higher Level Architecture
  • Project Start Architecture (DYA)?

• EA may prescribe views
  • Business, Information Systems, Technology

• Involve Enterprise Architects as Stakeholders
  • Briefings, workshops, reviews
  • Independent solution reviewers?