The Costing View of Architecture

Eltjo R. Poort
SATURN 2014

Experience the commitment®

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 Amsterdam, Twente, Eindhoven)
- Member if IFIP WG 2.10 Software Architecture

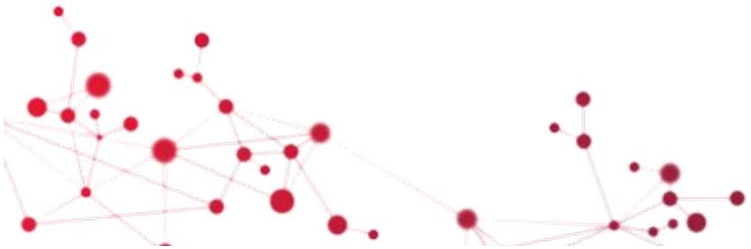
http://eltjopoort.blogspot.com

Impact of Architecture on Project Control

Quantified by research*

Result	Improvement by applying Solution Architecture	
Budget predictability	2-3 x better	Std dev 32 → 13
Budget overrun	7 x less	22% → 3%
Time overrun	6 x less	48% → 8%
Troubled projects	3 x less	38% → 13%
Customer satisfaction	1-2 points better	10 point scale
Results delivered	+10%	

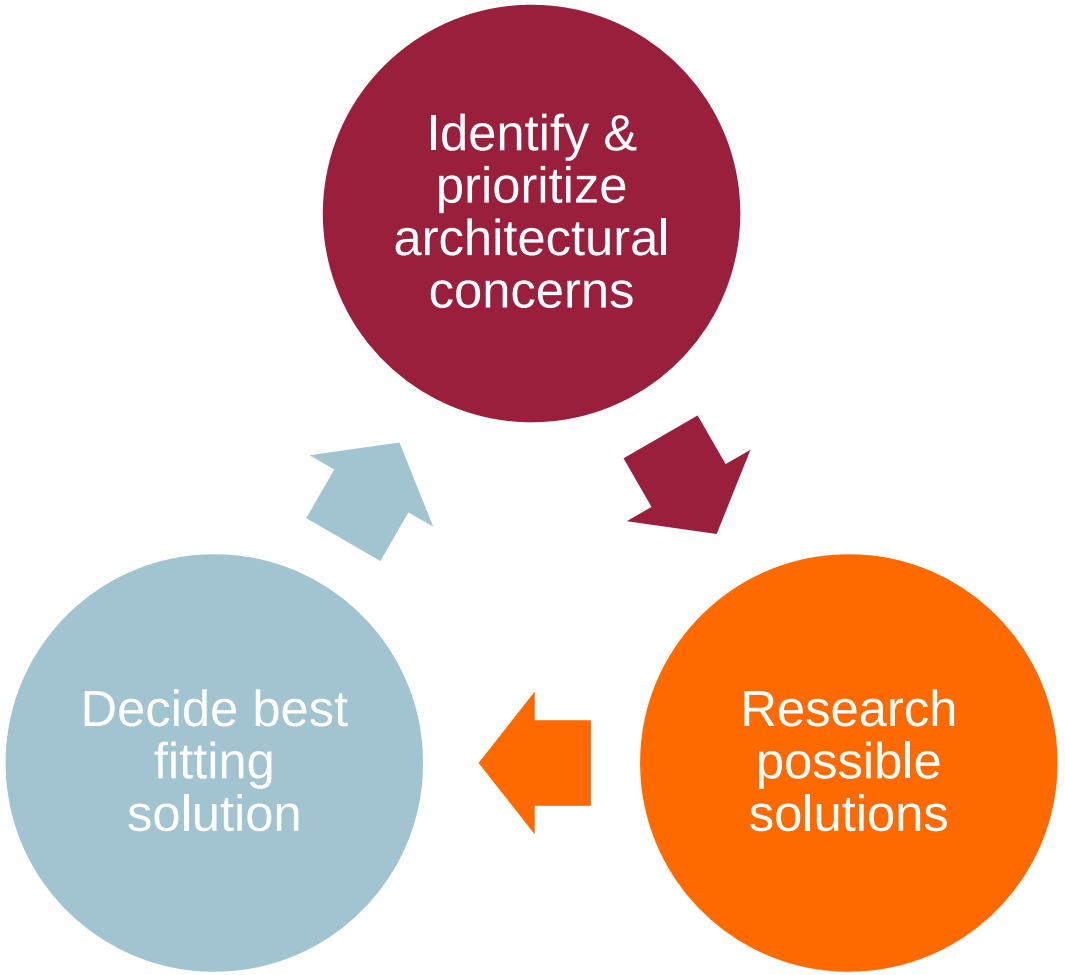
Specifically correlated with presence of architect and defined solution architecture during budget calculation



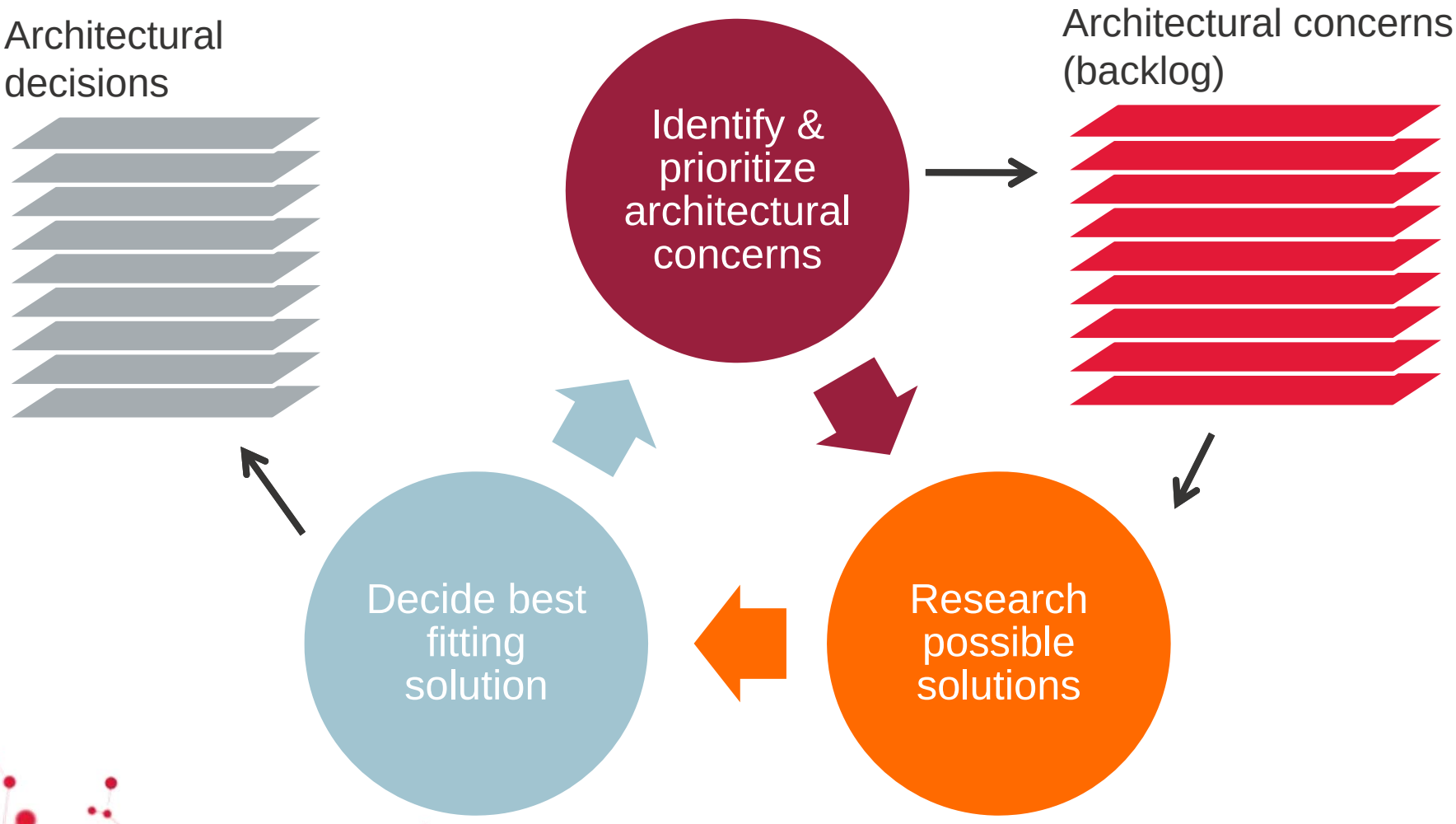
*Survey among 49 software development projects between €50,000 and €2,500,000. Reported by Raymond Slot, PhD Thesis, 2010.

CGI

The Architecting Microcycle



The Architecting Workflow



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".

[ISO/IEEE]

"Architecture is about the important stuff. Whatever that is."

[Fowler]

design

Editor: Martin Fowler ■ ThoughtWorks ■ fowler@acm.org

Who Needs an Architect?

Martin Fowler

Wandering down our corridor a while ago, I saw my colleague Dave Rice in a particularly grumpy mood. My brief question caused a violent statement, "We shouldn't interview anyone who has 'architect' on his resume." At first blush, this was an odd turn of phrase, because we usually introduce Dave as one of our leading architects.

The reason for his title schizophrenia is the fact that, even by our industry's standards, "architect" and "architecture" are terribly overloaded words. For many, the term "software architect" fits perfectly with the smug controlling image at the end of *Matrix Reloaded*. Yet even in firms that have the greatest contempt for that image, there's a vital role for the technical leadership that an architect such as Dave plays.

What is architecture?

When I was fretting over the title for *Patterns of Enterprise Application Architecture* (Addison-Wesley, 2002), everyone who reviewed it agreed that "architecture" belonged in the title. Yet we all felt uncomfortable defining the word. Because it was my book, I felt compelled to take a stab at defining it.

My first move was to avoid fuzziness by just letting my cynicism hang right out. In a sense, I define *architecture* as a word we use when we want to talk about design but want to puff it up to make it sound important. (Yes, you can imagine a similar phenomenon for architect.) However, as so often occurs, inside the blighted cynicism is a pinch of truth. Understanding came to me after reading a posting from Ralph Johnson on the Extreme Programming mailing list. It's so good I'll quote it all.

A previous posting said

> The RUP, working off the IEEE definition, defines architecture as "the highest level concept of a system in its environment. The architecture of a software system (at a given point in time) is its organization or structure of significant components interacting through interfaces, those components being composed of successively smaller components and interfaces."

Johnson responded:

> I was a reviewer on the IEEE standard that used that, and I argued uselessly that this was clearly a completely bogus definition. There is no highest level concept of a system. Customers have a different concept than developers. Customers do not care at all about the structure of significant components. So, perhaps an architecture is the highest level concept that developers have of a system in its environment. Let's forget the developers who just understand their little piece. Architecture is the highest level concept of the expert developers. 'What makes a component significant? It is significant because the expert developers say so.
>
> So, a better definition would be "In most successful software projects, the expert developers working on that project have a shared understanding of the

2 IEEE SOFTWARE Published by the IEEE Computer Society 0740-7459/03/$17.00 © 2003 IEEE

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.

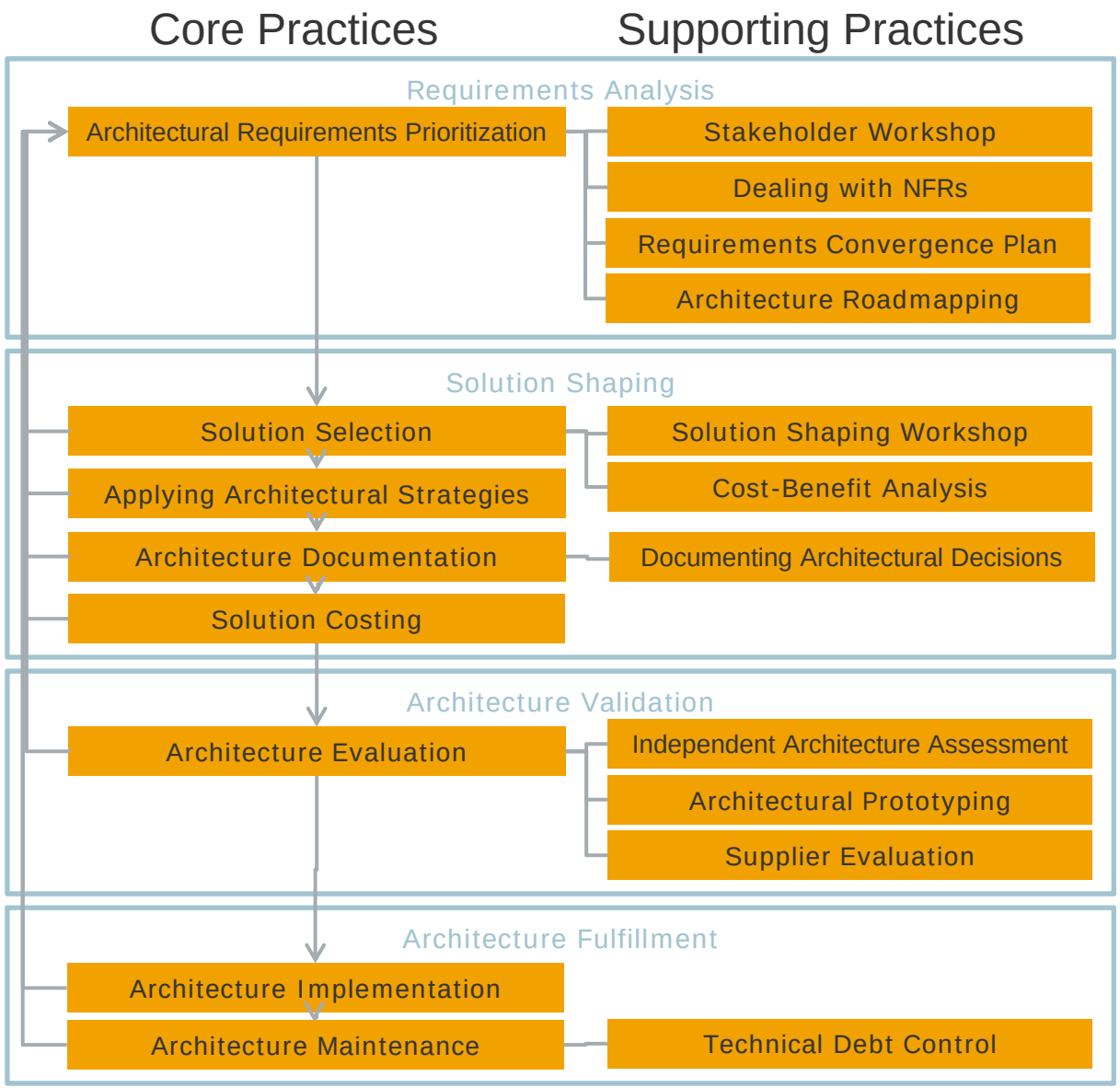
Important ←→ high risk and cost

Risk and Cost Driven Architecture

Solution architecting principles and practices based on a view of architecture as a risk and cost management discipline

- Applicable in agile and traditional engagements
- Highly scalable and pragmatic
- Architectural decision making based on economic trade-offs
- Architecture communication in economic terms
- Traceability from requirements to cost

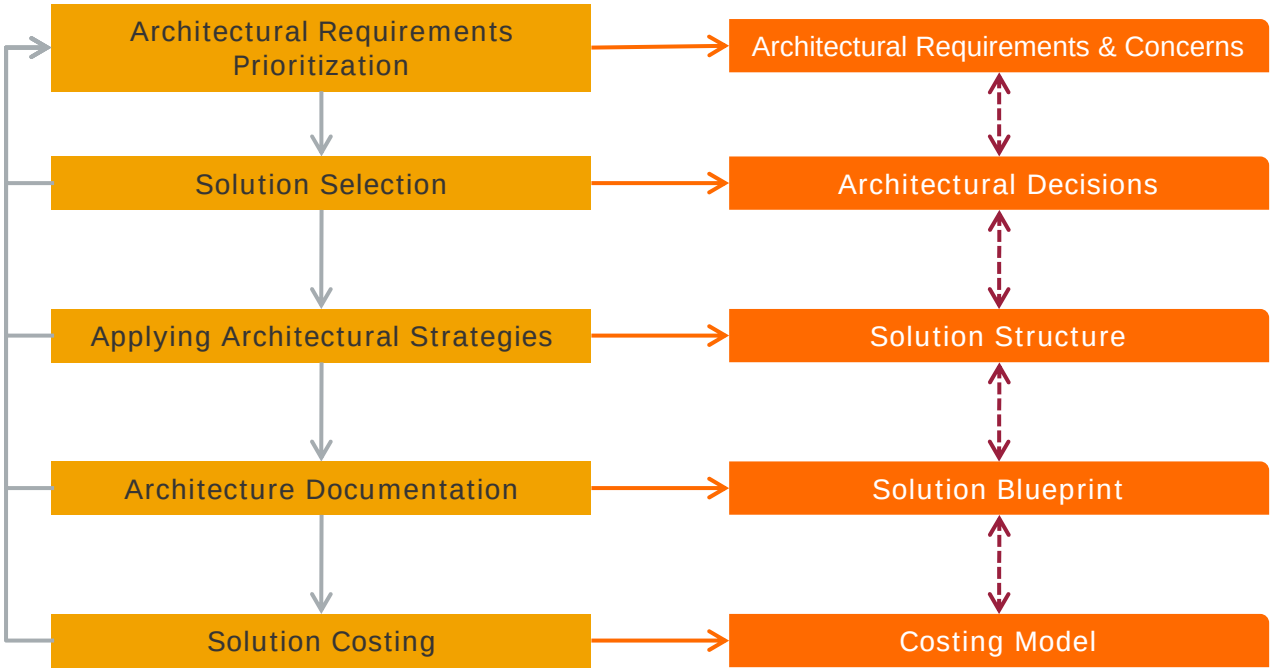
RCDA Practices

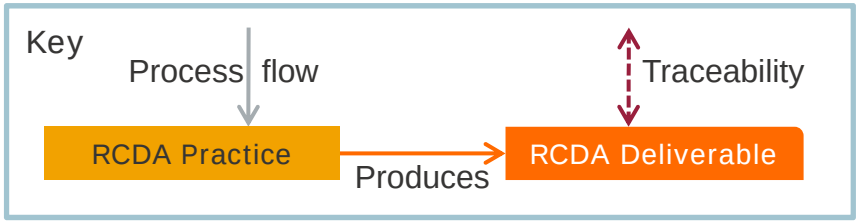


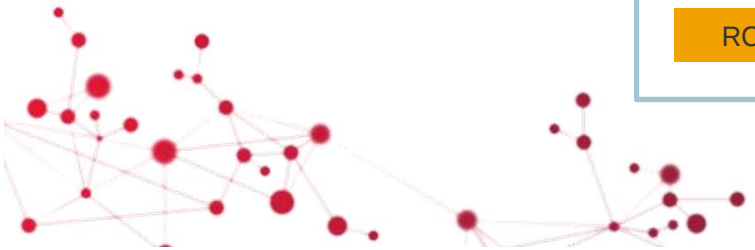
Lifecycles

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

Traceability from requirements to cost



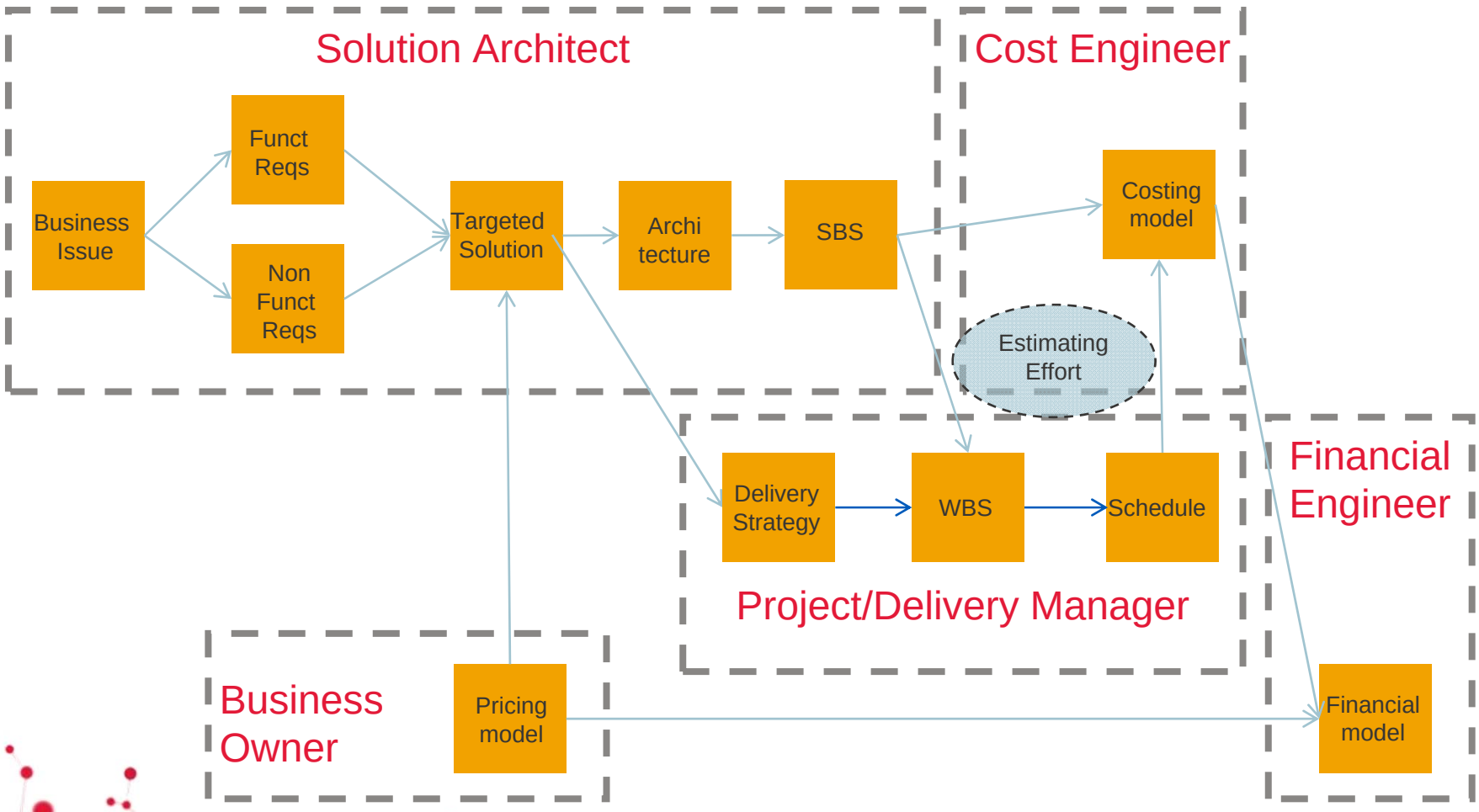




CGI

Solution-based estimating

Collaboration

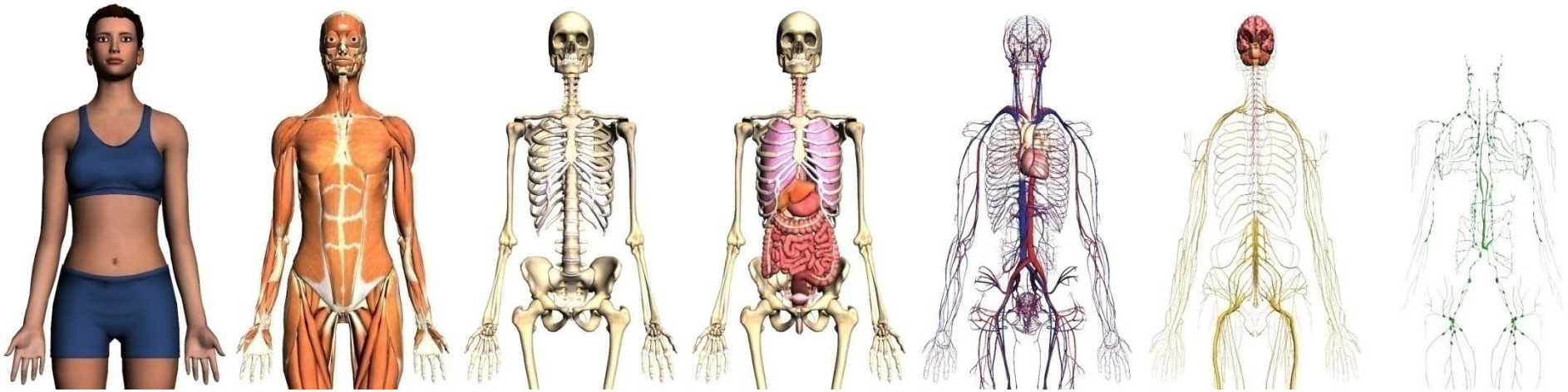


Architecture Documentation
Views and Viewpoints

All architecture documentation methods use views

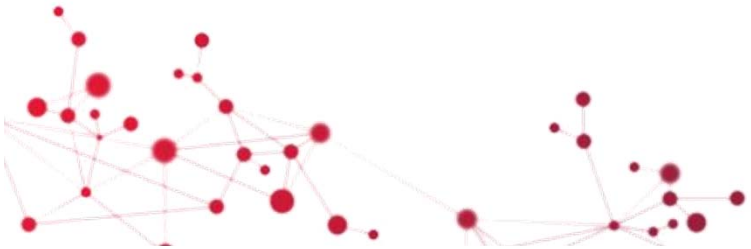
- ISO 42010, TOGAF, Archimate, 4 + 1, ‘Views and Beyond’

- Viewpoints address concerns per stakeholder (group)
- RCDA Solution Definition template has standard viewpoints
- Don’t forget: connect the views!



Solution Definition

Document Template for Planning/Budgeting/Bid stage



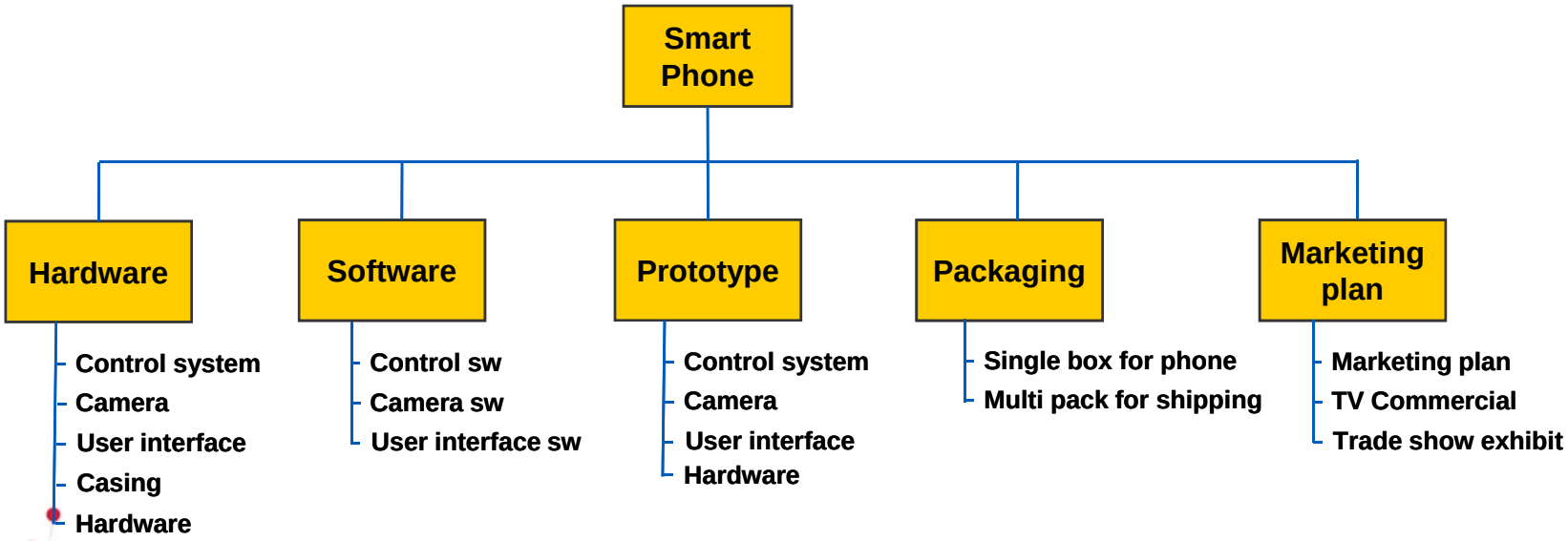
CGI

Solution Breakdown Structure (SBS)

The SBS is a deliverable-oriented hierarchical decomposition of the solution

The SBS is a tree showing how the solution decomposes into products

- top level of the tree is the Solution itself
- lower levels show how each product breaks down into sub-products, etc.

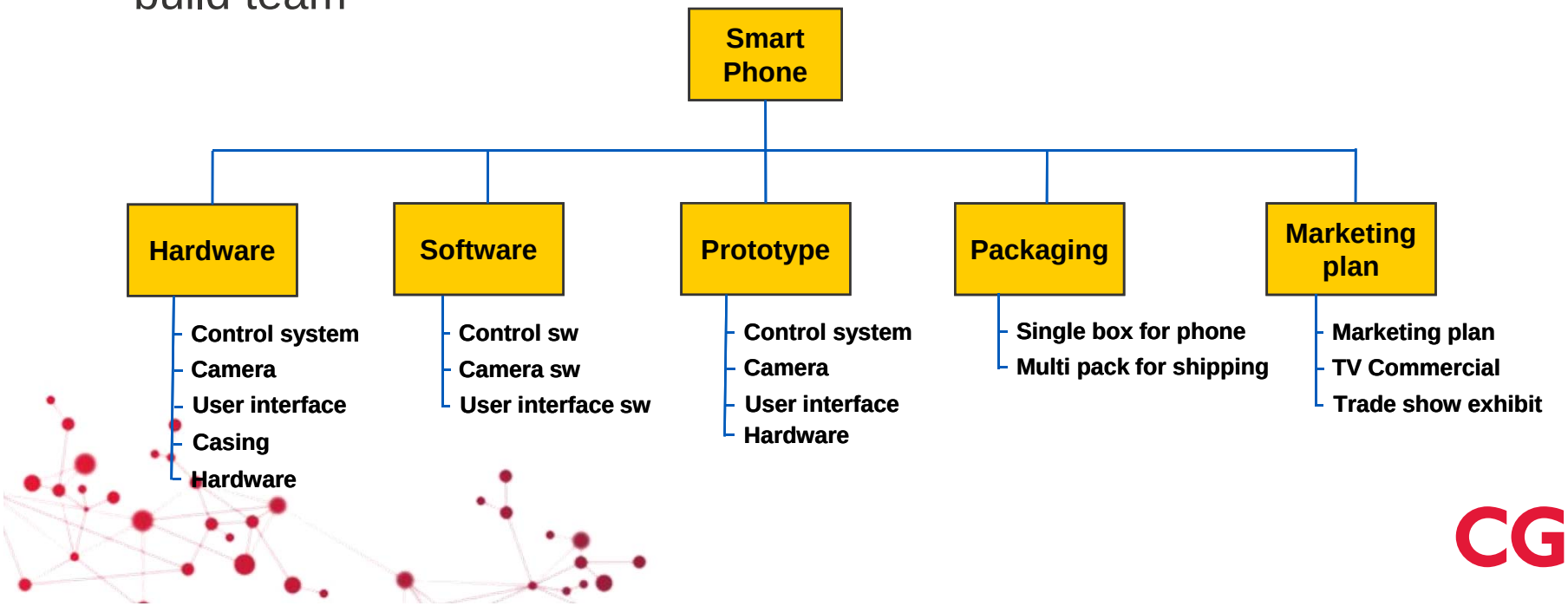


Solution Breakdown Structure (SBS)

Depth of the tree is determined by level of detail required for a reasonable cost estimate

At lowest level, each product should be responsibility of a single delivery organisation, e.g:

- service line
- single subcontractor
- build team

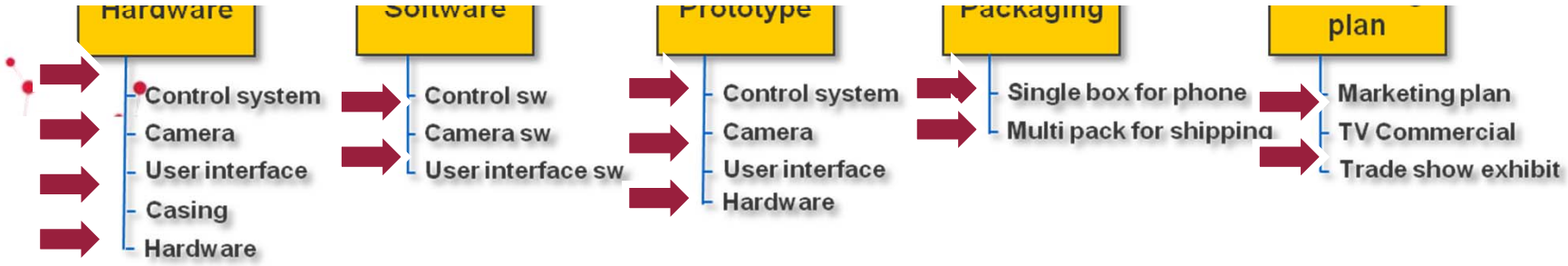


Development Approach

Development Approach: high-level activities required to develop or otherwise obtain the products that make up the solution

Responsibility for selecting delivery approach lies with Project/Delivery Manager

- but consult Solution Architect to make sure of proper fit with solution's structure and requirements

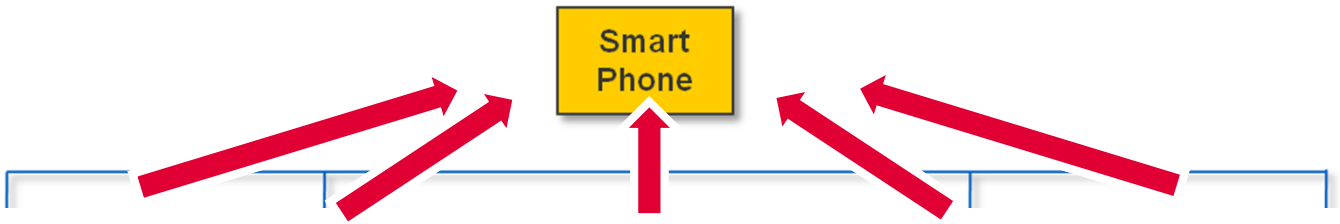


Integration Strategy

Integration Strategy: describes how the various products in the SBS are integrated to form the complete solution

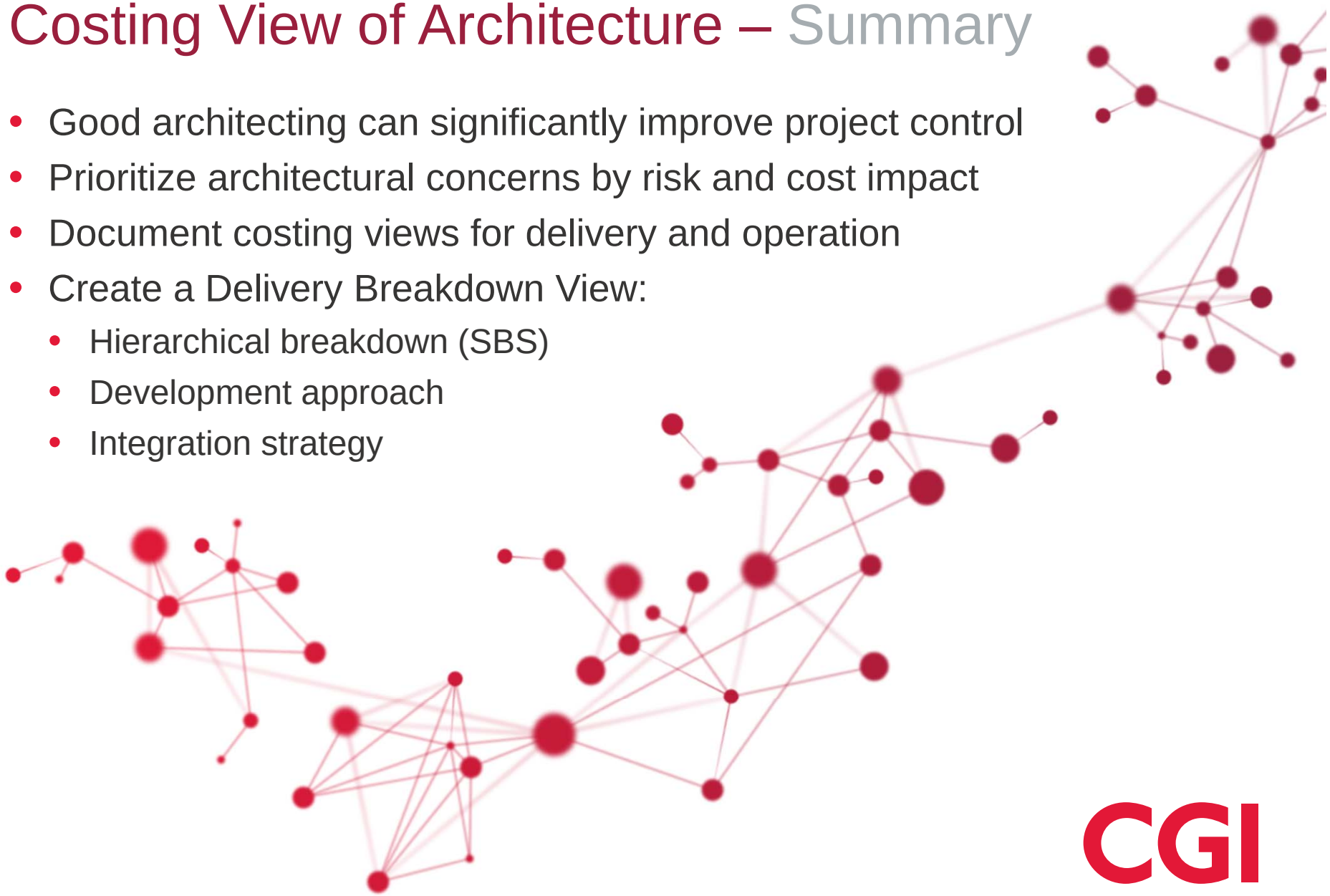
Responsibility for determining integration strategy lies with Project/Delivery Manager

- but consult Solution Architect to make sure all architectural concerns regarding the implementation are addressed



Costing View of Architecture – Summary

- Good architecting can significantly improve project control
- Prioritize architectural concerns by risk and cost impact
- Document costing views for delivery and operation
- Create a Delivery Breakdown View:
  - Hierarchical breakdown (SBS)
  - Development approach
  - Integration strategy



CGI

Experience the commitment®