



Architecting Agile Businesses:

A Guideline for the Business-Oriented Software Architect

Kaine Ugwu

SATURN 2016

Kaine Ugwu

Software Architect

Konga Online Shopping Ltd.

<http://kaine.pro>

- *Planning new technology insertion*
- *Assisting business in formulating clear requirements and making architectural tradeoffs*
- *Engaging engineering team during development, resolving disputes*
- *Defining, documenting and communicating the architecture*



Presentation Scope

- *Architecture-centric methods & patterns overview*
- *Recommended guidelines for architecting Agile businesses*
- *Benefits of adopting SOA patterns & ATAM style design peer reviews*
- *Lessons learned*



The background is a solid pink color. In the top right corner, there are several overlapping geometric shapes: a light pink triangle pointing down-left, a dark pink triangle pointing up-right, and a dark pink square. The text is centered on the left side of the slide.

*How do we
improve business
agility?*

Architecture-Centric Methods

- *Establishing Requirements - **Quality Attribute Workshop (QAW)***
 - *Defining an Architecture - **Attribute-Driven Design (ADD)***
 - *Evaluating the Architecture - **Architecture Tradeoff and Analysis Method (ATAM)***
 - *Documenting the Architecture - **SEI 's Views & Beyond Approach (V&B)***
- 

Quality Attributes

- *Non-functional Requirements*
- *Significant Influence on the Software Architecture*
- They are usually the **Architecturally Significant Requirements** that require the architects' attention

https://en.wikipedia.org/wiki/List_of_system_quality_attributes



Service-Oriented Architecture

“A service-oriented architecture (SOA) is an architectural pattern in computer software design in which application components provide services to other components via a communications protocol, typically over a network. The principles of service-orientation are independent of any vendor, product or technology”

- *Wikipedia*



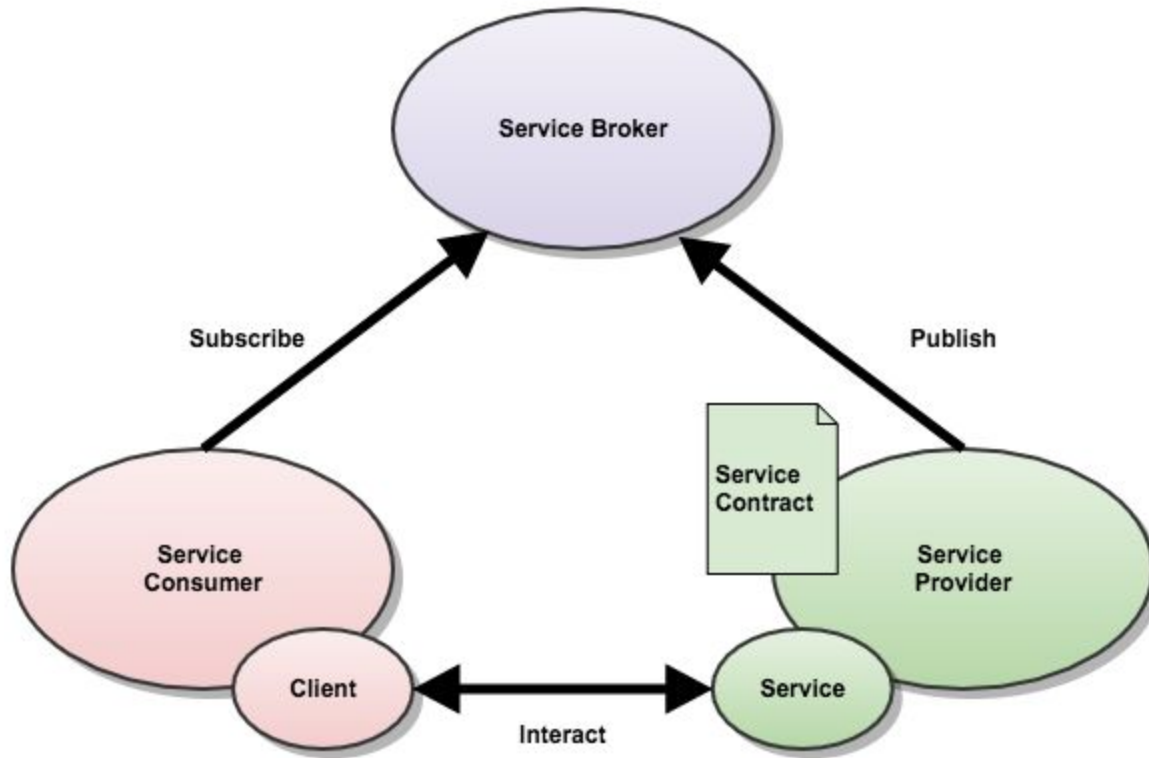


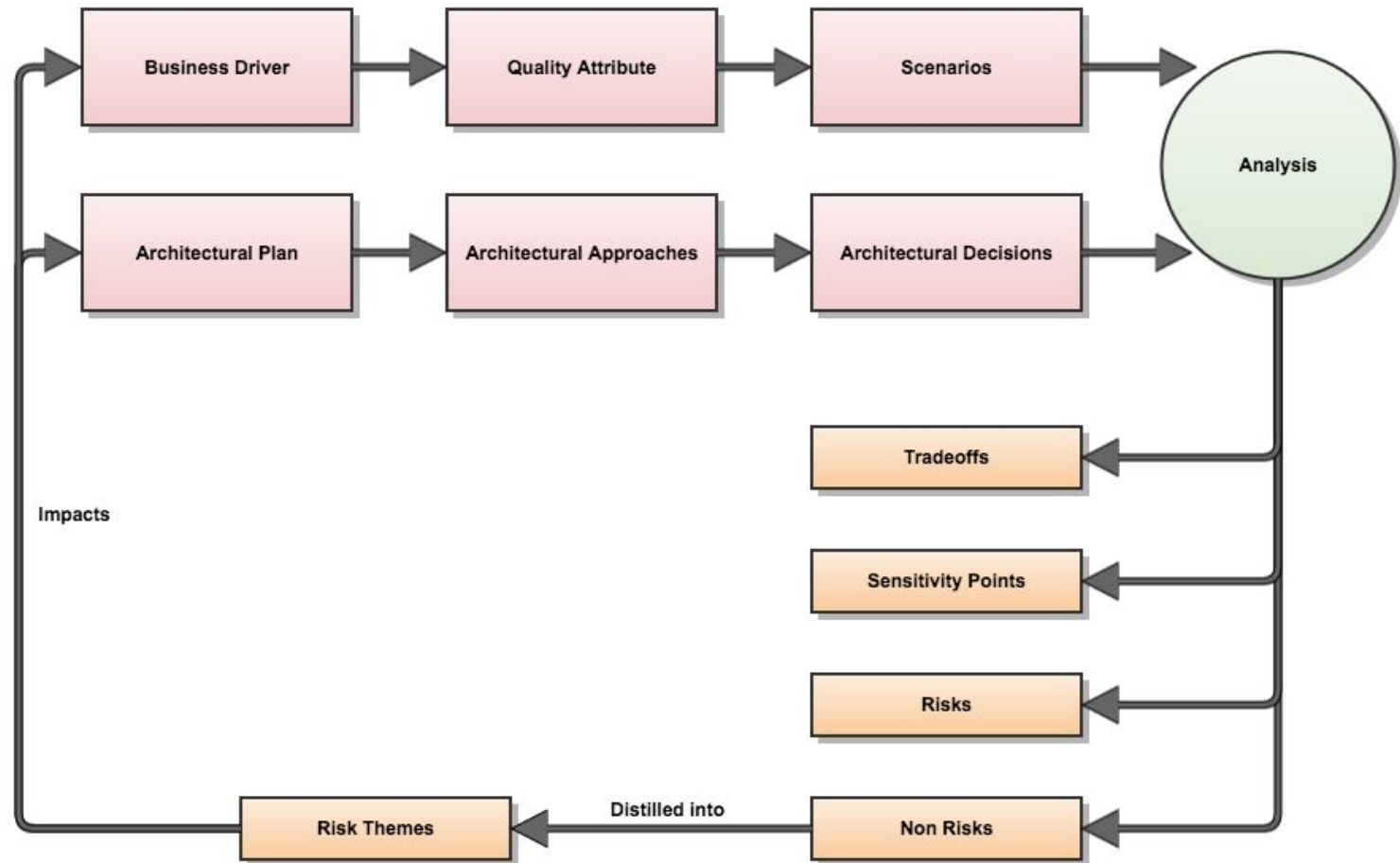
Figure: Service-Oriented Architecture Pattern

Architecture Tradeoff Analysis Method

*“The Architecture Tradeoff Analysis Method (ATAM) is a method for evaluating software architectures relative to **quality attribute** goals. ATAM evaluations expose **architectural risks** that potentially inhibit the achievement of an organization's business goals”*

- Software Engineering Institute





*Figure: ATAM Conceptual Flow
(Software Engineering Institute, CMU)*

Business People:

1. Don't like long technical processes.

2. Don't understand technical jargon.



We want our software architecture lifecycle processes to be...

- *Fast*
- *Iterative*
- *In a language business would understand*
- *Adhere to proven methods*
- *Get buy in from stakeholders*





*Recommended Guidelines for
Architecting Agile Businesses*

- *Represent Business Processes*
- *Service-Oriented Architecture Pattern*
- *ATAM Style Design Reviews*



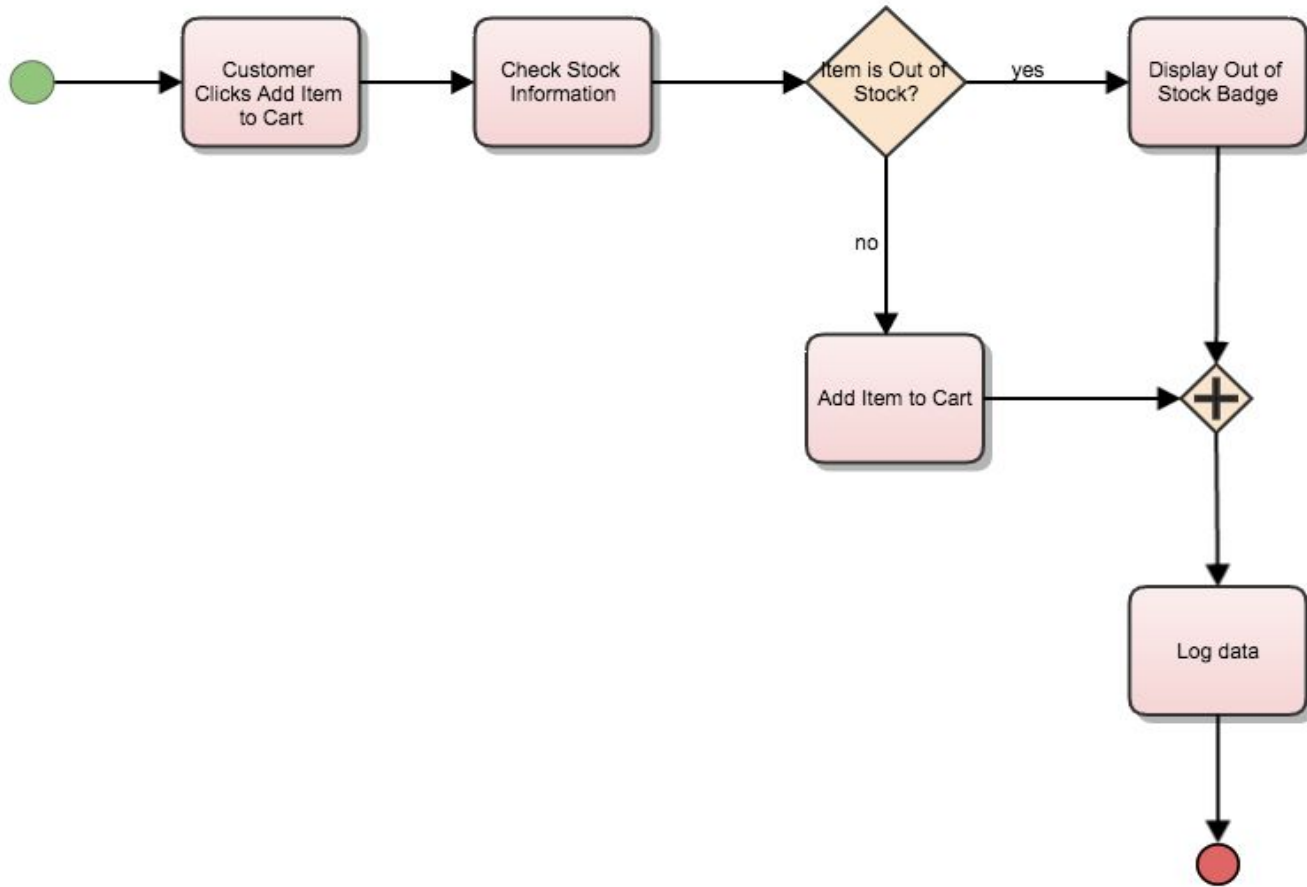
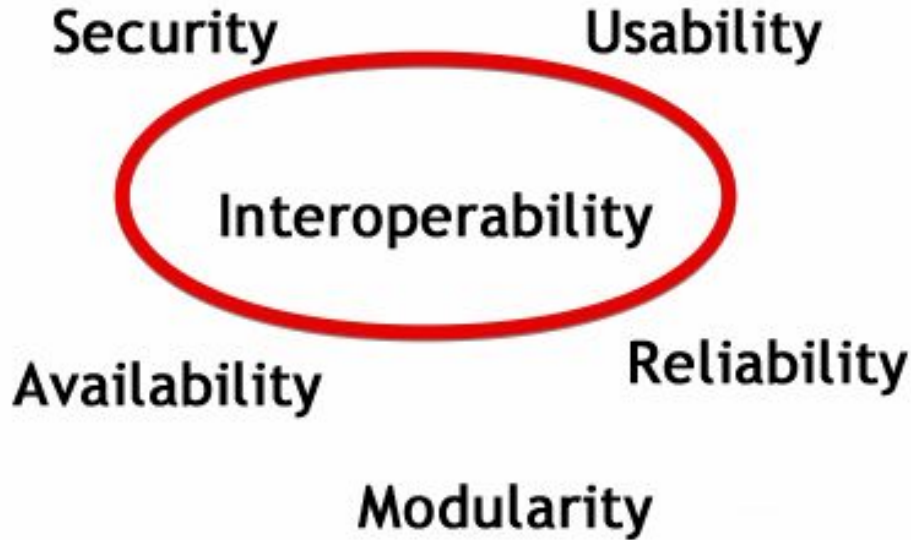


Figure: Shopping Cart Requirement Graphical Representation using BPMN

Step 1: Select the scenario to analyze.



Interoperability

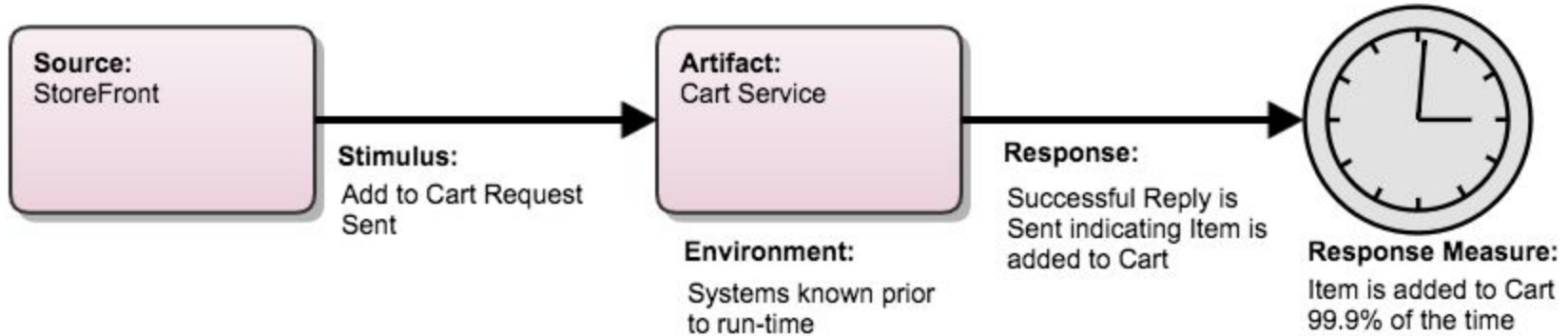


Figure: Interoperability Concrete Scenario
(Konga Shopping Cart)

Step 2: Elicit the architecture approaches

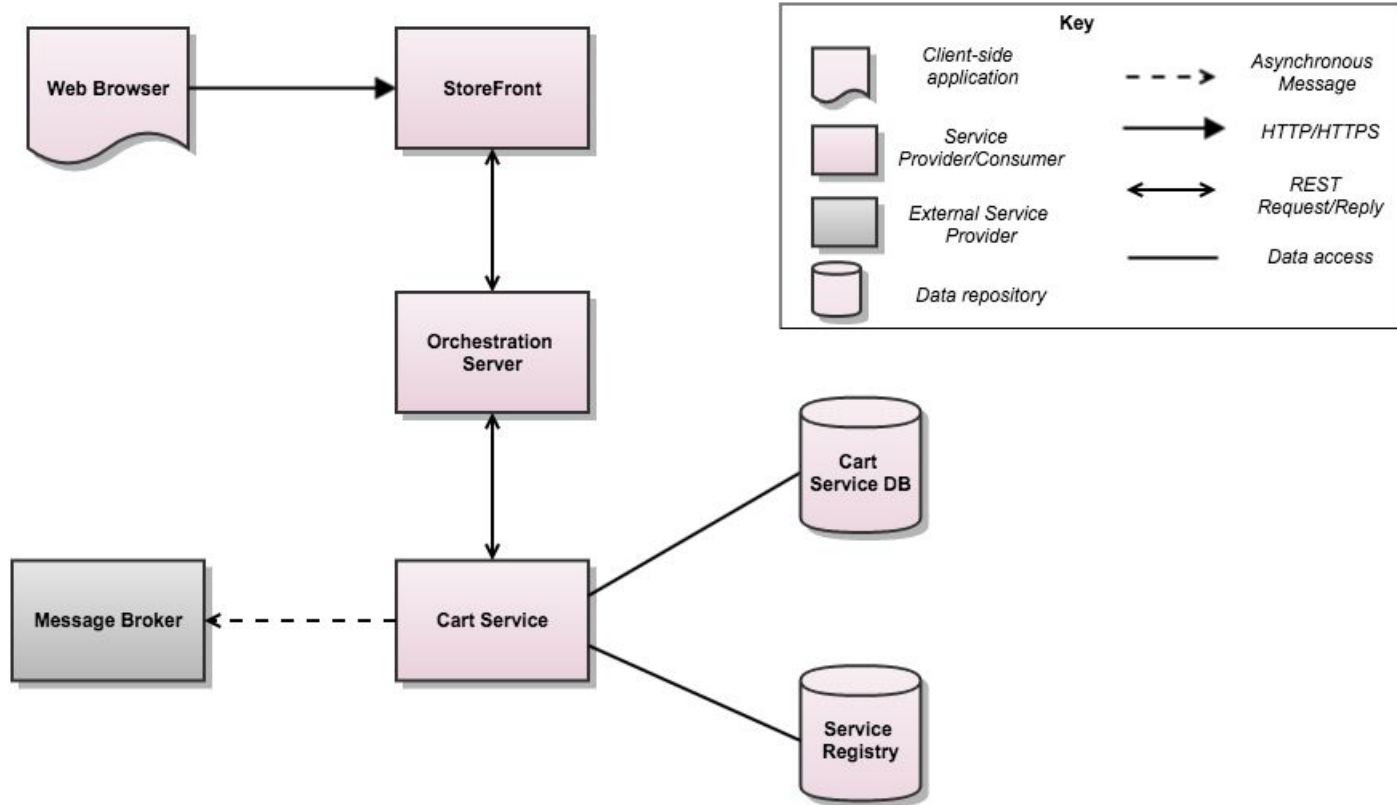


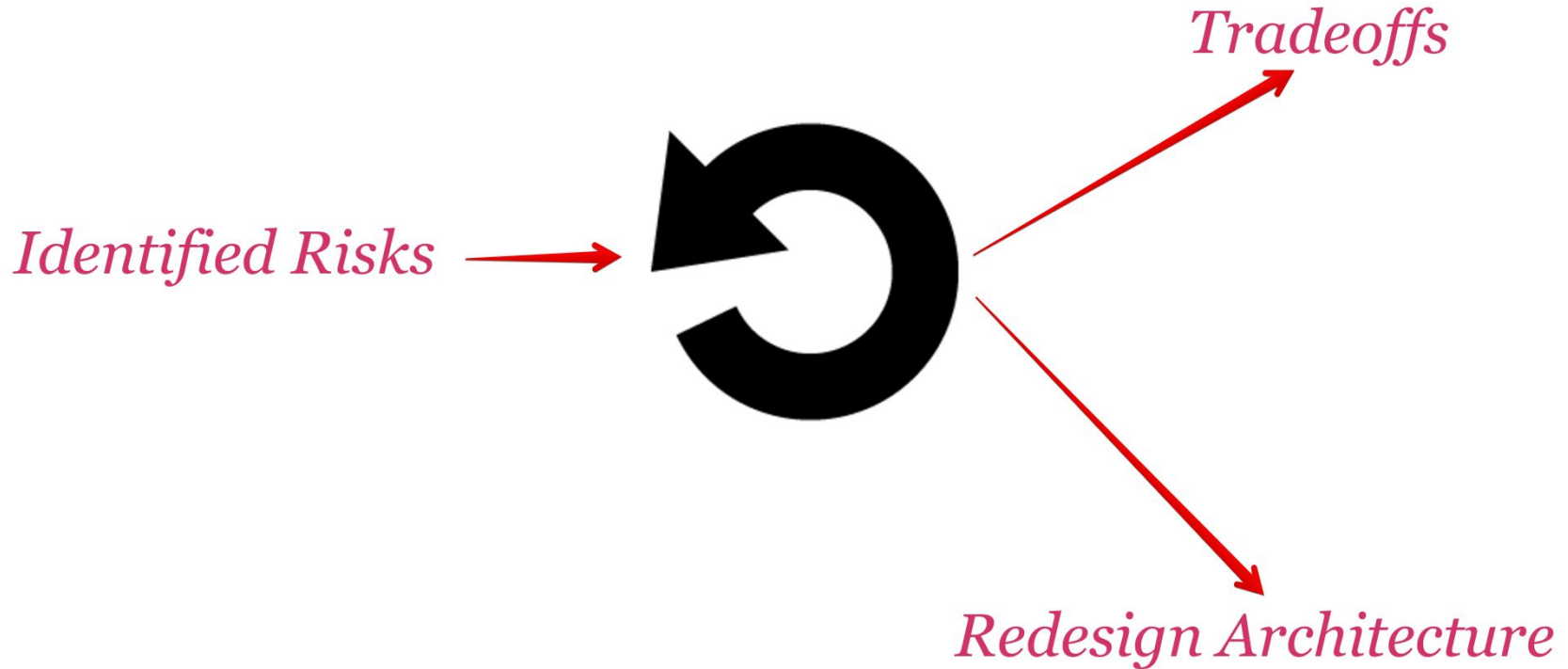
Figure: Diagram of the SOA view for the Konga Shopping Cart System

Step 3: Analyze architecture approaches

- If a question **cannot be answered**, it is identified as a **risk**
- If the provided answer **may violate other scenarios**, it is identified as a **risk**
- If the provided answer is still an **open issue**, it is identified as a **to-do item**
- If the provided answer **satisfies** the reviewers, it is documented as **evidence**



Step 4: Review results






*Benefits of Adopting SOA
Patterns & ATAM Style Design
Peer Reviews*

Benefits of SOA

- *Loose Coupling*
- *Service Re-use*
- *Higher Availability & Better Scalability*
- ***Ship software faster*** ← *What the business is really concerned about.*




Benefits of ATAM Style Design Reviews

- *Precise business drivers and quality requirements are gathered*
 - *Includes risk identification & management early in the life-cycle*
 - *Encourages communications among stakeholders*
 - *Conflicting goals are prioritized*
 - *Overall improved architecture practices*
 - ***Business and IT alignment***
- 



Lessons Learned

Lessons Learned

- *Business folks don't understand technical jargon, use **common business language**.*
 - ***Stakeholder sign-off** is extremely important.*
 - ***Service discovery** is extremely important*
 - ***Simplify** methods as much as you can*
- 



*Simplify methods
and patterns as
much as you can.*

*“Architecture is architecture is
architecture”*

- John Zachman

Thanks!

Kaine Ugwu

Software Architect

kaine@kaine.pro

kaine.ugwu@kongas.com

www.kaine.pro

[@kainepro](#)



Questions?