Architectural Engagement Through The Project Lifecycle

A Project / Programme Managers Guide

Daljit Banger  MSc FBCS
As a Project Manager (PM), I need to understand the Architectural resources available to my project..

Who are the Architecture Community?

What are the Typical Products of Architecture?

How do you derive value from the architecture community?

When should you utilise the resources available to you?
The Architecture Community
Enterprise Architects

At a minimum they

- Stakeholder Management
- Maintain and socialise the **macro** viewpoint of the technology ecosystem
- Map and maintain the business technology requirements and the associated capabilities.
- Managing Business requirements to support the current and future operating model/states of the organisation.
- Managing systems compliance to internal and external standards
- Manage the inventory of systems and components
- Managing Technical Debt
- Drive *technical value for money*

Carry out some of the following activities

- **Strategic** input into the technology roadmaps of the organisation to shape, form and stabilise where required.
- **Influence** decision makers on technology investment – current & future
- Provide systems **consultancy**, guidance, and assurance to large Programmes
- Understand the Business Value Streams, Capabilities etc and review and **assure** Solution Designs produced both internally and by 3rd party suppliers against the streams.
- Ensure that **governance** mechanisms, such as review boards, principles, etc. are maintained and supported and are part of project gates
- **Police** the standards through Project and Programme engagement
- Represent the organisation with **3rd parties**, for example Systems Integrators and Standards bodies
- Understand the **impact** of the introduction of new technology into the technology landscape of the organisation.
- Develop and promote a catalogue of reusable proven **patterns**
Solution Architects work with the projects and programmes with clearly articulated outcomes and are responsible for the delivery of designs, impact analysis, compliance to standards and support the needs of the project. At a minimum they can:

- Assist with technical problem Identification
- Cost Estimation for technical work packages
- Contextualise problems
- Eliciting the non-functional requirements
- Deliver the High-Level Solution Design
- Identify the Technology ‘Pick List’
- Work on the Route to Live
- Support transition into Service

Solution Architects “Drive the move between a Business/Technical Problem once identified to the delivery of a Systems Solution when applicable”
The Architecture Community

Technical Architect

- **Technical** Architects with Solution Architect to assist in the realization of the solution.
- TAs are a **key project resource** especially at the delivery stages of the project.
- They work with the technology enablers and deploy, manage, and support the running of the services required for the business to operate.

- Delivering **technical designs** and standards and the associated approvals from the formal governance channels.
- **Awareness and understanding** of the ‘as is’ technology estate and technology components deployed in the organisation.
- Providing technical **recommendations** and options based on solution designs which can cost-effectively be realised in the production environment.
- Mitigating any technical **risks** that could occur through the introduction of new technology into the landscape of the organisation.
- Providing input into the appropriate innovation funnels for the **analysis** of new technology.
- Keeping abreast of **technology trends**, attending industry events to ensure product roadmaps are understood by the Solution and Enterprise Architects.
- Ensuring that **production acceptance** for projects is delivered and managed.
- Performing **Impact assessments** on selected technology.

Operational / Build – Delivery View
Architect acts as a Technical Authority, to establish technical guard rails and drive establishment of an architecture runway with which the teams can apply design concepts to evolve their design incrementally within the guard rails.
• High Level Designs (HLDs)
  • Macro Level Design
  • Capability Maps
• Solution Designs (SD's)
  • How the Requirements will be met
  • Design with
    • Workflows
    • Component Lists
    • Work with Sprint Leads
• Technical Designs (LLDs)
  • Support (Project/Sprint)
Waterfall Touchpoints

Analysis
- Requirements
- Current State
- Future State
- Gap / Impact Analysis
- Do Ability
- Security

Design
- Solution (LLD)
- Use Cases
- Functional
- Non-Functional
- Components
- Integrations
- Data model
- Security Model
- Support
- Etc

Build
- Infrastructure
- Communication links
- Security Models
- Integrations
- Orchestrations

Test
- Software test Life Cycle
  - Static testing.
  - Unit testing.
  - Integration testing.
  - System testing.
  - Acceptance testing.

Deploy
- Deployment Architecture
- Environment Configuration Build
- Release Notes
- Promotion to Live

Run
- Runtime Monitor
- Fault / Performance
- SOC
- Support

Security / Service Architects
Agile Cycle Touchpoints

Idea → Value Stream → Features → User Story → Personas → Product Backlog → Execution → TA → Sprint Planning → TA → Sprint Reviews → Finished Products → Retrospectives

SA → TA → SA
## Estimation (Effort) Considerations

### Estimation Mix

<table>
<thead>
<tr>
<th>Opportunity</th>
<th>Delivery Style</th>
<th>Process Count</th>
<th>Project</th>
<th>Architectural Governance</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Build</td>
<td>• Waterfall</td>
<td>• New</td>
<td>• Cost</td>
<td>• Lightweight</td>
</tr>
<tr>
<td>• Buy</td>
<td>• Agile</td>
<td>• Refactored</td>
<td>• Effort</td>
<td>• Internal</td>
</tr>
<tr>
<td>• Refactor /</td>
<td></td>
<td></td>
<td>• Duration</td>
<td>• External</td>
</tr>
<tr>
<td>Configure</td>
<td></td>
<td></td>
<td></td>
<td>• Compliance</td>
</tr>
</tbody>
</table>

### Activity | Typical Outputs | Waterfall | Estimated Effort | Agile | Estimated Effort |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasibility</td>
<td>• Options Paper</td>
<td>Per Deliverable</td>
<td>Effort will be aligned to the project type – where small projects will require 'light weight' designs and arms length guardrail controls</td>
<td>Days</td>
<td>Weeks</td>
</tr>
<tr>
<td>Requirements Analysis / Traceability</td>
<td>• Capability Maps</td>
<td>Prep-Time / Presentation Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NFR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Product Backlog</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solution Design (HLD)</td>
<td>• Document</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Inventory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Support Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solution Walkthroughs</td>
<td>• Governance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Black Hats</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Acceptance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Selection / Backlog Definition</td>
<td>• Req Map</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• User Stories</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Pattern Usage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Tech Debt</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprints / Build</td>
<td>Definition of Number of Sprints</td>
<td>Per Stage N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing</td>
<td>• Static testing.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Unit.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Integration..</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• System.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Acceptance.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production Acceptance</td>
<td>Release Notes</td>
<td>Subject to Scale</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLD / Security</td>
<td></td>
<td>(L) – 2-3 Weeks (S) – 4-5 Days</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The effort will map directly to the sprint activity, there are however some activities which will be cross cutting and will require multiple sprints e.g. integration, performance modelling, security – the non functional stuff.
What are the challenges, you have faced, when working with the Architecture Community?

Open Discussion....
URLs

- SFIA - The global skills and competency framework for the digital world - https://sfia-online.org/
- Architectural Services / Touch Points - A Project / Programme Managers Guide This presentation ..
  - The Blog https://dalbanger.blogspot.com/