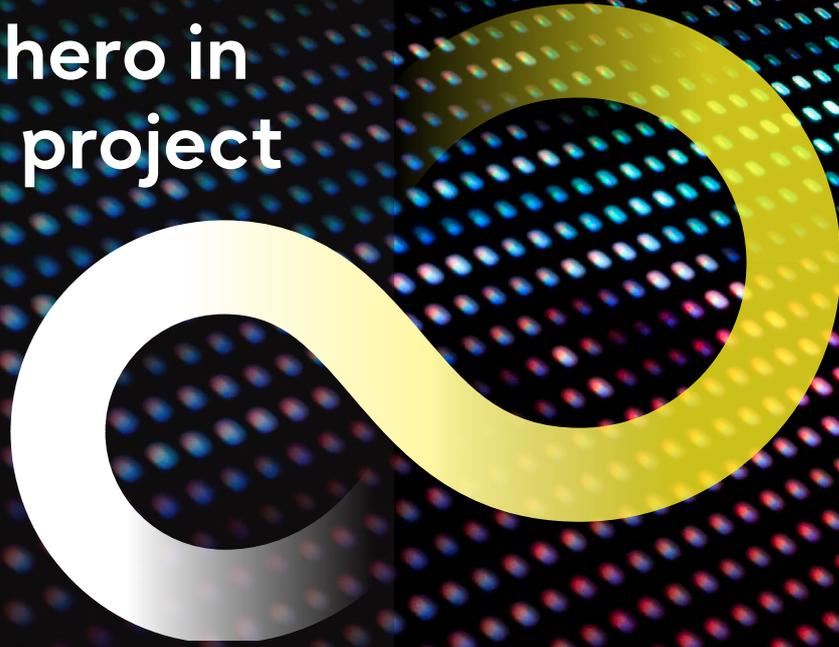


Pragmatic architecture – the quiet hero in your next project



All IT projects use architects. Sometimes the role is given other names – Technical Lead, Senior Developer, Project Manager – but no matter what you call it, you need an architect to design, govern, and drive your work to ensure it is robust, fit-for-purpose, and enables outcomes. Good architecture provides the critical foundations of a delivery; it should never *determine* the outcome; but it should be pragmatic and *enable* the outcome.

Whatever the mission statement of the project, the architecture must enable and deliver on the promise. If your architect produces cutting-edge gold standard designs, but they don't enable the desired outcomes, are they really gold standard?

Pragmatic architecture, whether greenfield or BAU, needs to align with MVP principles - cost, time, quality, and scope are primary considerations.

Similarly, architectural design must comply with business requirements. But should it massively exceed them? No! Over-engineering a solution adds no value and only serves to blow out the core MVP principles. Pragmatic architecture should aim to slightly over-deliver on the current release and be built with sufficient flexibility so that it can grow as the need arises for the continually evolving releases.

I am sure we have all come across the 'evangelist architect'. Evangelists architect completely innovative, and yet completely impractical, solutions. They are driven by pride, prestige, and reputation and have less interest in business outcomes. But the role of a pragmatic architect is more challenging. They must filter the needs and wants of many stakeholders with technological, financial, and environmental considerations. They must make critical governance-level decisions on patterns, technologies, and approach. And most importantly, they must ensure their solution aligns with the business' transformation needs. While they may not attract the same amount of attention as the evangelist, the pragmatic architect is the quiet hero in the story.

So how do we become more pragmatic? An emerging industry term, 'composable architecture', is gaining ground. This concept is entirely founded on the core Object Oriented Programming Principles - polymorphism, encapsulation, data abstraction, and inheritance. These programming principles enforce simplicity; they ensure code is re-usable and extensible and limits unnecessary re-work. But the principles align just as well with pragmatic architecture as they do with programming. Lightweight, connectable, diverse, and re-usable architecture modelling drives the ability to change gears or change direction as required without having to re-architect the complete solution.

Pragmatic architecture is the primary foundation for project success. Choose your architect carefully. Insist on composable architecture patterns. And ensure your architect understands your roadmap and transformation goals. Architecture should never be carved in stone. Pragmatic architecture is, and should forever be, pliable, mouldable, and responsive to the needs of business.



Tips to designing pragmatic architecture:

- **Architect to MVP + 1** - Any designed solution must meet the needs of MVP + 1 phase of delivery. Architecture chasing requirements will deliver haphazard outcomes; pragmatic architecture aligns with the roadmap of delivery and always keeps horizon n+1 in view.
- **Architecture or yoga?** - Loosely coupled, composable architecture will keep your deliveries lean, agile, and flexible. Aim for lithe architectures.
- **Challenge the architect** - Your architect may be the quiet hero behind your project, but you must challenge them. Your insights might change their world view, and their insights may influence yours.
- **Avoid FIFO architecture** - The more an architect understands the business, the user base, and the roadmap for delivery, the greater the return will be in produced patterns and fit-for-purpose architectures. This cannot be achieved with a fly-in fly-out approach. The pragmatic architect is your best friend; keep them close.
- **ArchOps** - Architecture, just like development, security, operations, and core aspects of business needs to 'shift left'. Fail fast, fail early. Design, design, design until it's right and synthesizable for the n+1 release. Architecture often remains the last vestige of waterfall delivery; it should be revisited, tuned, and maintained for the life of the architecture.