

What They Didn't Tell You About Architecture in Agile Development

Sriram Narayan, ThoughtWorks

Scope

Will talk about

- Underlying philosophy

- IT vendors, IT consumers, ISVs

- Enterprise, solution/system, application

- People skills

- Some examples

Won't talk about

- Architecture Frameworks

Relevance? Are you kidding?

Architecture is definitely relevant.

Question is

"Are traditional full time roles of architect relevant?"

The responsibilities traditionally associated with these roles still exist.

Except that these responsibilities are now executed by very capable developers.

Underlying Philosophy

Evolutionary (adaptive over predictive)

Code is Design

Fail Fast

Just in time (last responsible moment)

Essence over ceremony

End to end principle, consistency vs. nimbleness

Value disciplines

Anti-patterns of enterprise architecture

Architecture as a shared service

Shared build and deploy team

Shared frameworks teams

Layered app dev teams (Conway's law)

Org wide coding conventions

Org wide codebase metrics, dashboard

Turning measurements into targets

Prescriptive tech stack

Industrial ethic (versus: autonomy, mastery, purpose)

People, soft skills

Economic Realism

Decentralized decision making

Need influencing ability

Ability to communicate at different levels of abstraction

Ask for forgiveness, not for permission

Humility

Some Recommendations

Autonomous full-life cycle teams

Be a custodian of

IT asset value

Non-functional requirements (cross-functional?)

Encapsulation

Service orientation

Technical Debt

Write code for at least 20% of your time

Focus on goals and principles without being prescriptive

Random examples

Domain modeling - hbm files

What open source libraries to use

Package structure

JDepend

Restful or soapy