

Software (Systems) Architecture Foundations

Lecture #7

Architect and Architecture Work

Alar Raabe

Recap of Last Lecture

The architect must be a prophet – if he can't see at least ten years ahead don't call him an architect.

Frank Lloyd Wright

- A software product line is a set of
 - software systems that **share a common, managed set of features** and that are developed from a common set of core assets in a prescribed way
 - **reusable assets** (called core assets) **based on a common architecture** and the software elements that populate that architecture
- Selected variation mechanisms must support
 - the variations reflected in the products (often manifested as different quality attributes)
 - the production strategy and production constraints (support the way the organization plans to build products)
 - efficient integration (a large number of products requires a smooth and easy process) – some degree of automation
- Primary architectural variation mechanisms are
 - *Inclusion or omission* of elements or inclusion of a different number of replicated elements
 - *Selection/substitution* of different versions of elements with the same interface but different behavioral or quality attribute characteristics (libraries or addons/plugins)
 - *Reflection* (ability to adjust the behavior based on the context)

Content

The life of a software architect is a long and rapid succession of sub-optimal design decisions taken partly in the dark

P. Kruchten

- Role of Architect
- Architecture work
 - Architecture process
 - Architecture governance
- Architecture in the context of agile development
 - Scaled Agile Framework (SAFe)
 - Disciplined Agile (DA)
- Conclusions

From Thicket to Garden



Role of Architect (TOGAF)

Architect = chief scapegoat & cat herder

IBM SF Chief Architect

- Architect has overall responsibility for ensuring
 - the **completeness** (fitness-for-purpose) of the architecture, in terms of adequately addressing all the concerns of its stakeholders
 - the **integrity** of the architecture, in terms of connecting all the various views to each other, satisfactorily reconciling the conflicting concerns of different stakeholders, and showing the trade-offs made in doing so
- The choice of which particular architecture views to develop is one of the key decisions that the architect has to make
 - **know and concentrate on the critical few details that really matter, and do not to become overloaded with the rest !**
- Responsibilities of architect
 - **Understand and interpret requirements** – facilitate consensus building, synthesize and translate ideas into actionable requirements, and articulate those to others
 - Create and maintain useful models
 - Validate, refine, and expand the models
 - Manage the architecture
 - **Continuously foster the sharing of information**

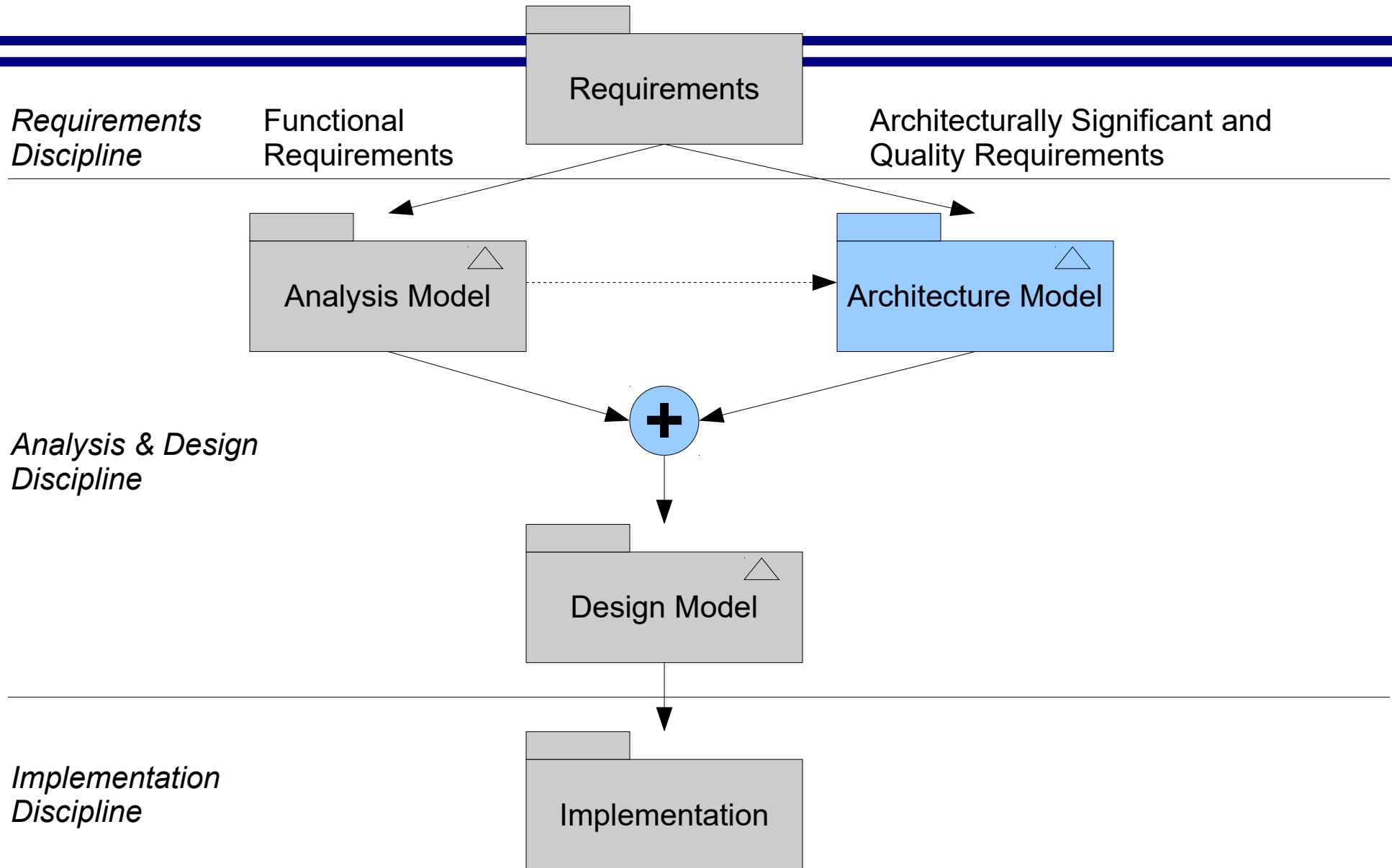
Content

The life of a software architect is a long and rapid succession of sub-optimal design decisions taken partly in the dark

P. Kruchten

- Role of Architect
- Architecture work
 - Architecture process
 - Architecture governance
- Architecture in the context of agile development
 - Scaled Agile Framework (SAFe)
 - Disciplined Agile (DA)
- Conclusions

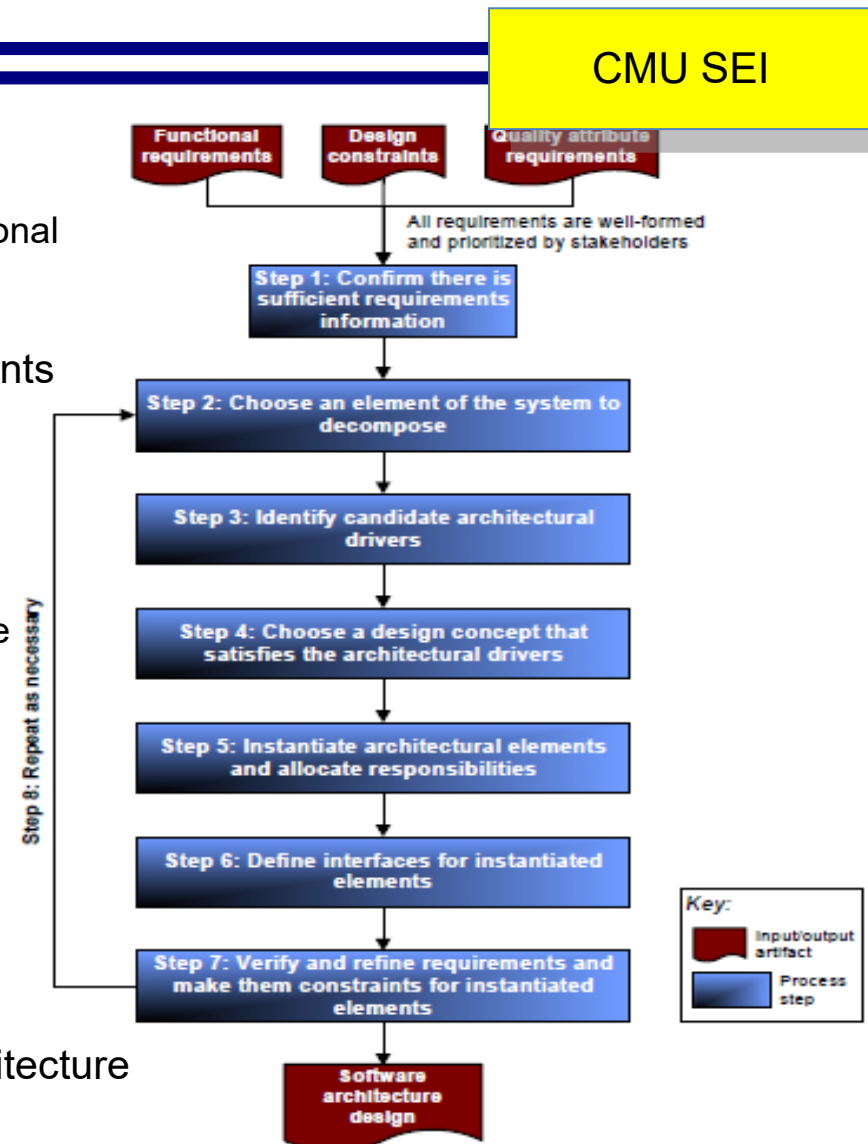
Creating Architecture



Activities of “Architecture Process”

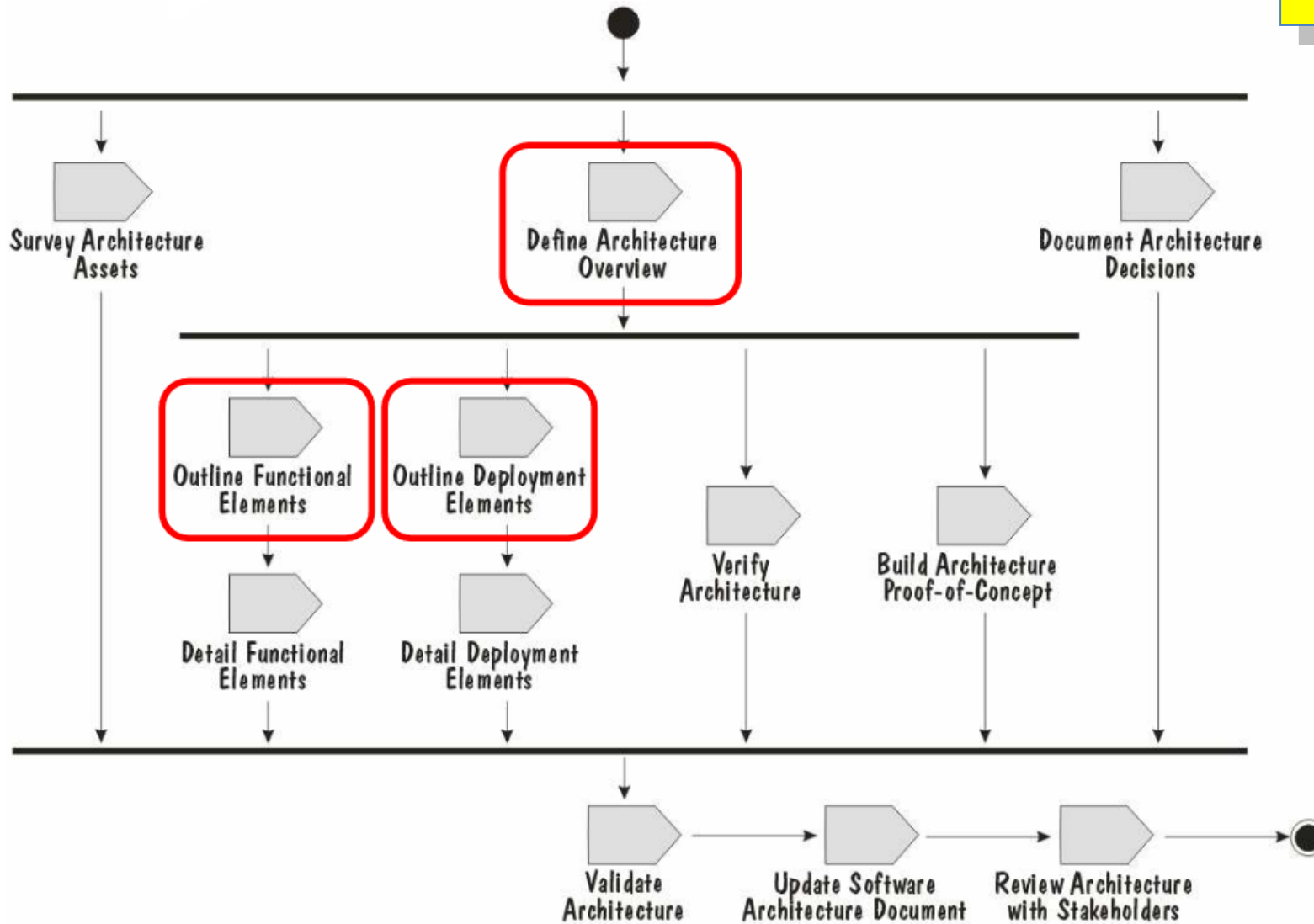
CMU SEI

- Making a business case for the system
 - A business case is, briefly, a justification of an organizational investment
- Understanding the architecturally significant requirements
- Creating or selecting the architecture
- Documenting and communicating the architecture
 - For the architecture to be effective as the backbone of the project's design, it must be communicated clearly and unambiguously to all of the stakeholders
- Analyzing or evaluating the architecture
- Implementing and testing the system based on the architecture
- Ensuring that the implementation conforms to the architecture



Creating Architecture

IBM RUP



Architecture Governance

TOGAF

- Governance is the system by which business corporations are directed and controlled (OECD, 2011)
 - Governance is essentially about ensuring that business is conducted properly, to ensure sustainability of an organization's strategic objectives
 - Characterized by
 - *Discipline* – commitment to adhere to procedures, processes, and authority structures
 - *Transparency* – availability for inspection of all actions implemented and their decision support
 - *Independence* – avoiding potential conflicts of interest in all processes and decision-making
 - *Accountability* – all who take actions or make decisions are obliged to inform and justify these
 - *Responsibility* – obligation to act for the benefit of the organization and its stakeholders
 - *Fairness* – impartial and just behavior in decisions taken, processes used, and their implementation
- Architecture governance is the practice and orientation by which enterprise architectures and other architectures are managed and controlled at an enterprise-wide level
 - typically does not operate in isolation, but within a hierarchy of governance structures (e.g. corporate governance, technology governance, IT governance, ...)

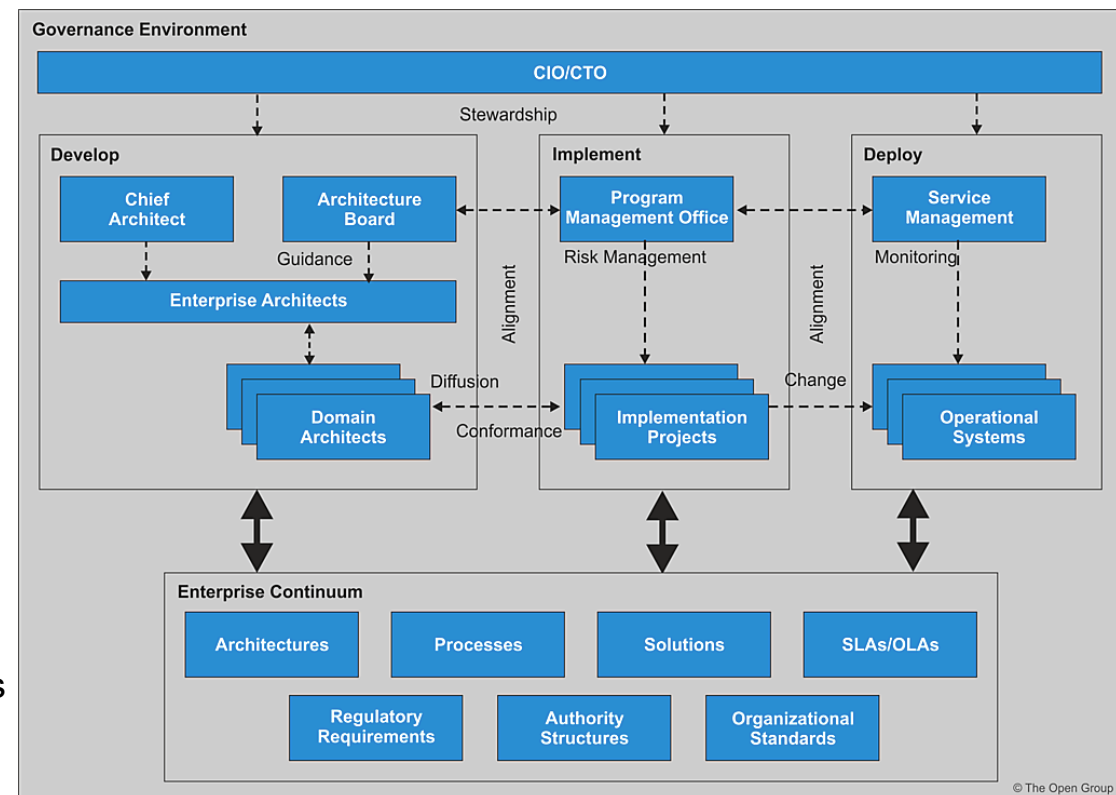
Architecture Governance Organization

TOGAF

- A key element in a successful Architecture Governance is a cross-organization **Architecture Board**
 - representative of all the key stakeholders in the architecture, and typically comprising a group of executives responsible for the review and maintenance of the overall architecture

- Roles

- Enterprise Architect
 - responsible at whole landscape and technical reference model level
 - focus on enterprise-level business functions required
- Segment/Domain Architect
 - responsible of specific business problems, domain or organizations
 - focus on enterprise-level business solutions in a given domain
- Solution Architect
 - responsible of systems or sub-systems
 - focus on system technology solutions



Content

The life of a software architect is a long and rapid succession of sub-optimal design decisions taken partly in the dark

P. Kruchten

- Role of Architect
- Architecture work
 - Architecture process
 - Architecture governance
- Architecture in the context of agile development
 - Scaled Agile Framework (SAFe)
 - Disciplined Agile (DA)
- Conclusions

Agile Architecture (P. Kruchten)

The best architectures, requirements, and designs **emerge** from self-organizing teams

Agile Manifesto

- **Agile Architecture** is either

- A system or software **architecture that is versatile, easy to evolve, to modify, flexible** in a way, while still resilient to changes

OR

- An **agile way to define an architecture**, using an iterative life-cycle, allowing the architectural design to tactically evolve gradually, as the problem and the constraints are better understood

- This naïve thinking about the spontaneous emergence of architecture is reinforced by the fact that **most software endeavors nowadays do not require a significant amount of new architectural design!**
- Architectural design, when it is really needed because of the project novelty, has an uneasy relationship with the traditional agile practices, because, **unlike functionality of the system, it cannot easily be decomposed in small little chunks of work**, user stories or “technical stories”

Agile Architecture

- A **Walking Skeleton** is a tiny implementation of the system that performs a small end-to-end function; it need not use the final architecture, but it should link together the main architectural components so that the architecture and the functionality can then evolve in parallel (A. Cockburn)
 - different from a spike, “the smallest implementation that demonstrates plausible technical success.” which typically takes between a few hours and a few days to complete, and is thrown away afterward, since it was built with non-production coding habits
 - **permanent code, built with production coding habits**, regression tests, and is intended to grow with the system
 - once the system is up and running, it will stay up and running for the rest of the project, despite the Incremental Re-architecture that is quite likely to occur

Architect Role in Agile

If architecture is the important stuff, then the architect is the person who worries about the important stuff

Martin Fowler

- *Architectus Reloadus* (after “Matrix Reloaded”)
 - person who makes all the important decisions, because a single mind is needed to ensure a system’s conceptual integrity, and perhaps because the architect doesn’t think that the team members are sufficiently skilled to make those decisions
- *Architectus Oryzus* (after D. Rice, “Patterns of Enterprise Application Architecture”)
 - person who is very aware of what’s going on in the project, looking out for important issues and tackling them before they become a serious problem (programs with a developer, trying to harvest some common locking code and participates in a requirements session, helping to explain the technical consequences of some of their ideas in non-technical terms — such as development costs)

Architect’s value (in agile development) is inversely proportional to the number of decisions he or she makes

Agile Architecture (SAFe)

While we must acknowledge emergence in design and system development, a little planning can avoid much waste

J. O. Coplien

- Agile architecture enables incremental value delivery by balancing between **emergent design** and **intentional architecture**
 - **emergent design** – provides the technical basis for a fully evolutionary and incremental implementation approach, which helps designers respond to immediate user needs, allowing the design to emerge as the system is built and deployed
 - **intentional architecture** – a set of purposeful, planned architectural initiatives, which enhance solution design, performance, and usability and provide guidance for inter-team design and implementation synchronization

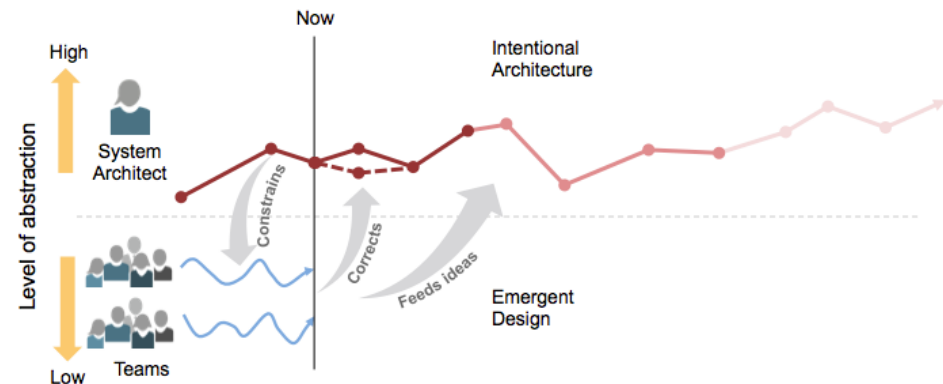
emergent design provides speed of change and intentional architecture ensures that the system as a whole has conceptual integrity and produces the intended result
- **Architectural Runway**
 - consists of the existing code, components, and technical infrastructure needed to implement near-term features without excessive redesign and delay
 - provides the necessary technical foundation for developing business initiatives and implementing new Features and/or Capabilities

Architect/Engineering (SAFe)

Architecture is a Collaboration

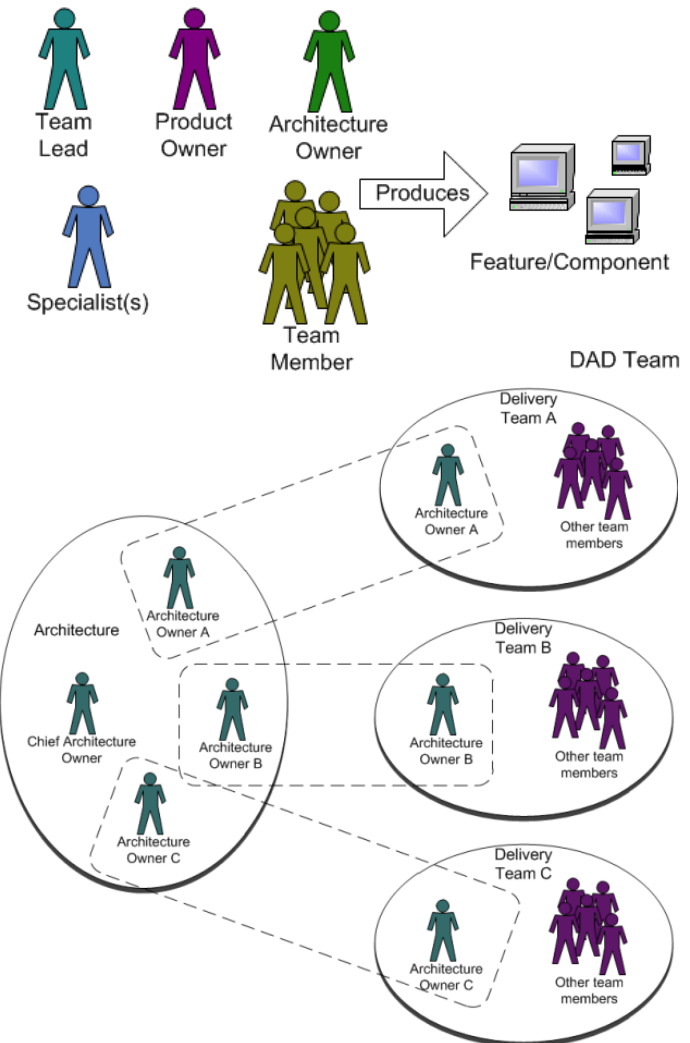
SAFe

- The **Solution Architect** role (an individual or small team)
 - takes a 'systems view' and helps to ensure fitness for purpose
 - defines a shared technical and architectural vision
 - participates in determining the system, subsystems, and interfaces, validate technology assumptions and evaluate alternatives, in definition of the higher-level functional and Nonfunctional Requirements (NFRs), analyzing technical trade-offs
- While providing a general view of solution architecture, Architect enables those who implement value by empowering them to make local decisions, allowing a faster flow of work and better economics
- Architects work at the program and large solution levels, often extending beyond the software domain to include responsibilities that enable value delivery in a technically diverse and heterogeneous, multi-domain solution environment
- The **Enterprise Architect** promotes adaptive design, and engineering practices and drives architectural initiatives for the portfolio
 - facilitate the reuse of ideas, components, services, and proven patterns across various solutions in a portfolio



Architecture Owner Role (Disciplined Agile)

- Architecture Owner is responsible for the team's architecture – just like Product Owner is responsible for the requirements
 - should have a technical background and a solid understanding of the business domain – often the most technically experienced person on the team
 - secondary role – not a hierarchical position into which other team members report
- Architecture Owner is needed because
 - architecture is a key source of project risks and someone needs to be responsible for ensuring mitigation of these
- Architecture Owner responsibilities
 - facilitating creation of the architecture, not enforcing it – an architecture owner collaboratively works with the team to develop and evolve the architecture
 - transitioning architectural skills to other team members
 - doing Architectural Spikes (a technical risk-reduction technique from XP where you write just enough code to explore the use of an unfamiliar technology or technique)
 - mentoring team members in organizational technical guidance
 - breaking decision "deadlocks"
 - reviewing architectural work from other teams



Content

The life of a software architect is a long and rapid succession of sub-optimal design decisions taken partly in the dark

P. Kruchten

- Role of Architect
- Architecture work
 - Architecture process
 - Architecture governance
- Architecture in the context of agile development
 - Scaled Agile Framework (SAFe)
 - Disciplined Agile (DA)
- Conclusions

Conclusions

If a builder builds a house for someone, and does not construct it properly, and the house falls and kills its owner, then that builder shall be put to death

Hammurabi, King of Babylon (1780 BC)

- Architect has overall responsibility for ensuring
 - the **completeness** and fitness-for-purpose of the architecture, in terms of adequately addressing all the concerns of its stakeholders
 - the **integrity** of the architecture, satisfactorily reconciling the conflicting concerns of different stakeholders, and showing the trade-offs made in doing so
- A key element in a successful Architecture Governance is a cross-organization **Architecture Board**
 - representative of all the key stakeholders in the architecture, and typically comprising a group of executives responsible for the review and maintenance of the overall architecture
- Architect role in
 - traditional development is to **design the architecture** – to make the decisions
 - agile development is to **enable/support/facilitate team to design the architecture** – not to make the decisions

17. The architect doesn't talk, he acts.
When this is done, the team says,
"Amazing: we did it, all by ourselves!"

Lao Tsu (by Philippe Kruchten)

Thank You!

Questions

- What is the role of architect?
- What are the main responsibilities of an architect?
- List the main activities of an “architecture process”?
- Why architecture governance is needed?
- How architecture capability could be set up in the enterprise?
- What is the role of architecture board?
- What is “agile architecture”?
- How architecture work is done in agile software development?
- Describe the role of architect in agile software development?
- What is “walking skeleton”?
- What is “architectural runaway”?

Literature

- <http://www.bredemeyer.com/who.htm>
- <http://epf.eclipse.org/wikis/openup/>
- https://resources.sei.cmu.edu/asset_files/TechnicalReport/2006_005_001_14795.pdf
- <http://pubs.opengroup.org/architecture/togaf9-doc/arch/toc-pt6.html>
- <https://philippe.kruchten.com/2013/12/11/agile-architecture/>
- <http://files.catwell.info/misc/mirror/2003-martin-fowler-who-needs-an-architect.pdf>
- <http://alistair.cockburn.us/Walking+skeleton>
- <http://www.scaledagileframework.com/system-and-solution-architect-engineering/>
- <http://agilemodeling.com/essays/architectureOwner.htm>
- <http://www.disciplinedagiledelivery.com/the-dad-role-of-architecture-owner/>
- <https://martinfowler.com/articles/ea-in-lean-enterprise.html>
-
- <https://www.mitre.org/sites/default/files/publications/se-guide-book-interactive.pdf>
-
- ... Google “software architect” + “agile architect” ...
-
- https://docuri.com/download/how-to-be-a-great-enterprise-architect-_59c1e09cf581710b2869b223_pdfa

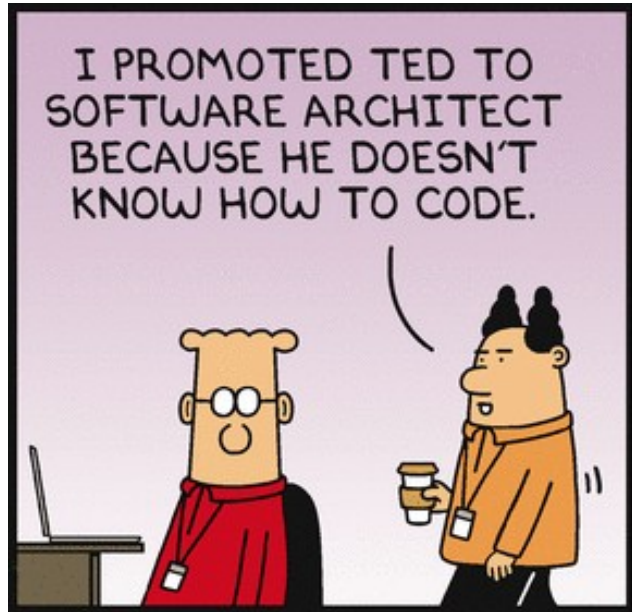
Example

Architecture Skills Framework

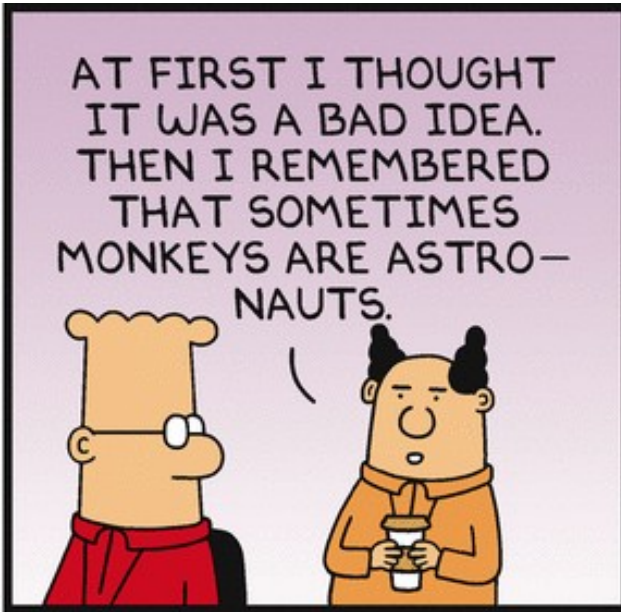
TOGAF

Roles		Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/Project Manager	IT Designer
Enterprise Architecture Skills										
	Business Modeling	2	2	4	3	3	4	4	2	2
	Business Process Design	1	1	4	3	3	4	4	2	2
	Role Design	2	2	4	3	3	4	4	2	2
	Organization Design	2	2	4	3	3	4	4	2	2
	Data Design	1	1	3	3	4	3	3	2	3
	Application Design	1	1	3	3	3	4	3	2	3
	Systems Integration	1	1	4	4	3	3	3	2	2
	IT Industry Standards	1	1	4	4	4	4	3	2	3
	Services Design	2	2	4	4	3	4	3	2	2
	Architecture Principles Design	2	2	4	4	4	4	4	2	2
	Architecture Views & Viewpoints Design	2	2	4	4	4	4	4	2	2
	Building Block Design	1	1	4	4	4	4	4	2	3
	Solutions Modeling	1	1	4	4	4	4	4	2	3
	Benefits Analysis	2	2	4	4	4	4	4	4	2
1	Background	3	3	4	3	3	4	4	3	1
2	Awareness	1	1	4	4	4	4	3	3	2
3	Knowledge	1	1	3	3	3	3	3	4	2
4	Expert	Extensive and substantial practical experience and applied knowledge on the subject.								

© The Open Group



Dilbert.com @ScottAdamsSays



7-18-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel

