



| IBM Software Group

The Rise of the Development Environment Architect

Peter Eeles

Chief Architect for IT

IBM Rational Worldwide Solution Delivery

peter.eeles@uk.ibm.com

Rational software

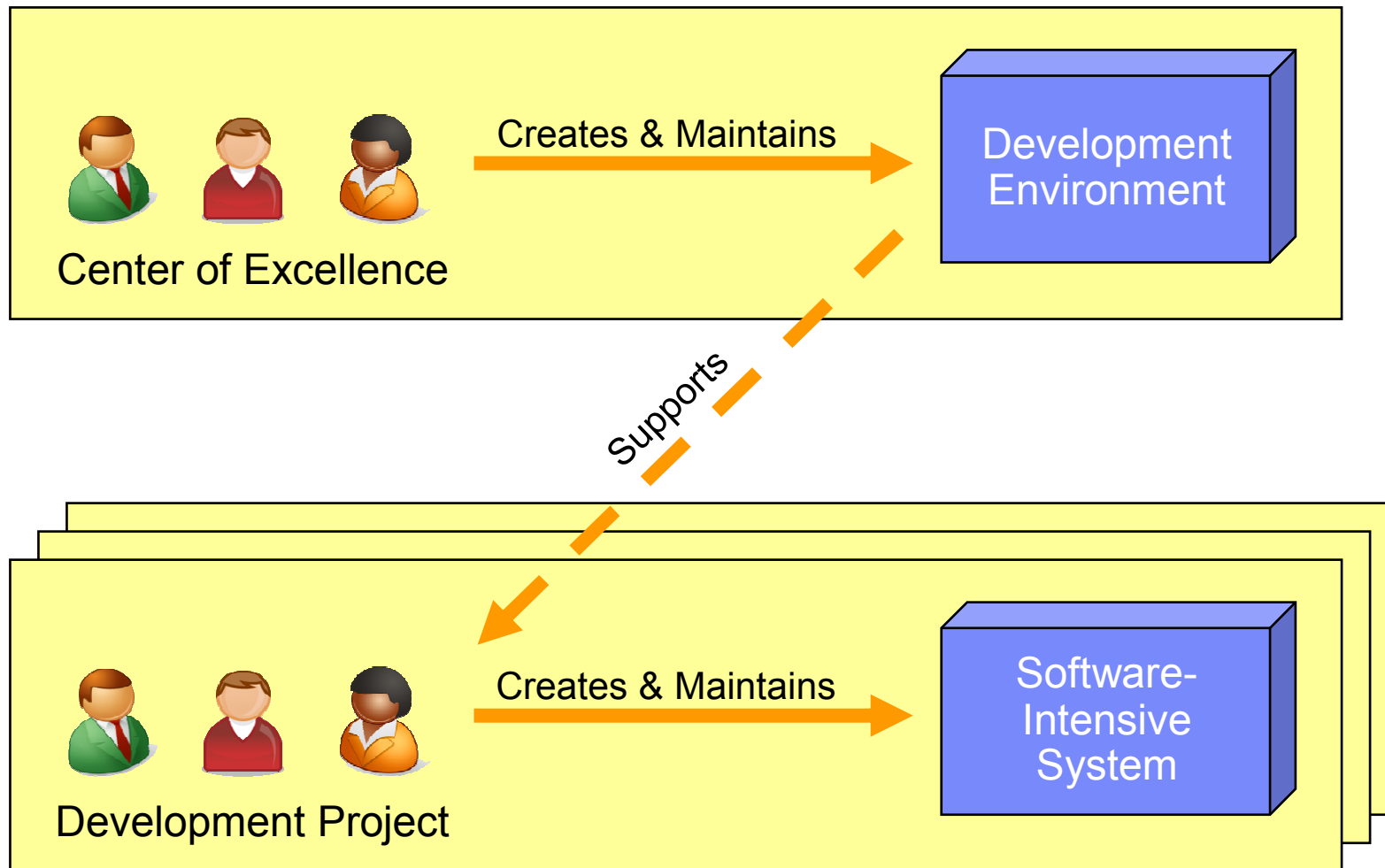


Agenda

- Start
- ➔ Development environment slides
- Architect slides
- Development environment architect slides
- Finish



Putting a Development Environment in Context

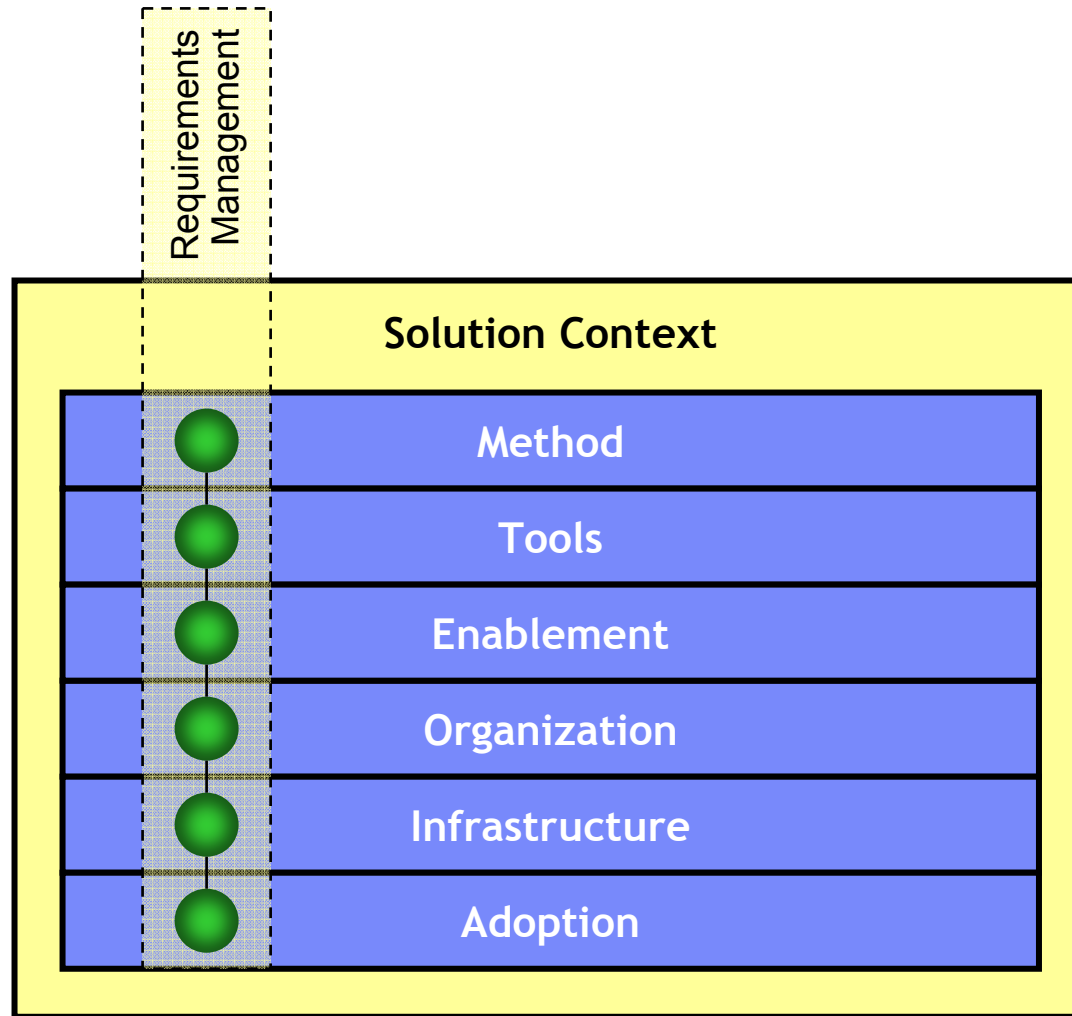


A Simple Definition

Solution Context (functionality, qualities, constraints)	
Method	Roles, work products, tasks, processes Standards, guidelines, checklists etc. Method deployment topology
Tools	Development tools and integrations Development tool configurations and install scripts Development tool deployment topology
Enablement	Training curriculum and courses Mentoring materials Enablement deployment topology
Organization	Organizational roles and units Organization deployment topology
Infrastructure	Locations, nodes and connectivity Supporting software (such as operating systems)
Adoption	Adoption plan Techniques for driving organizational change Environment metrics

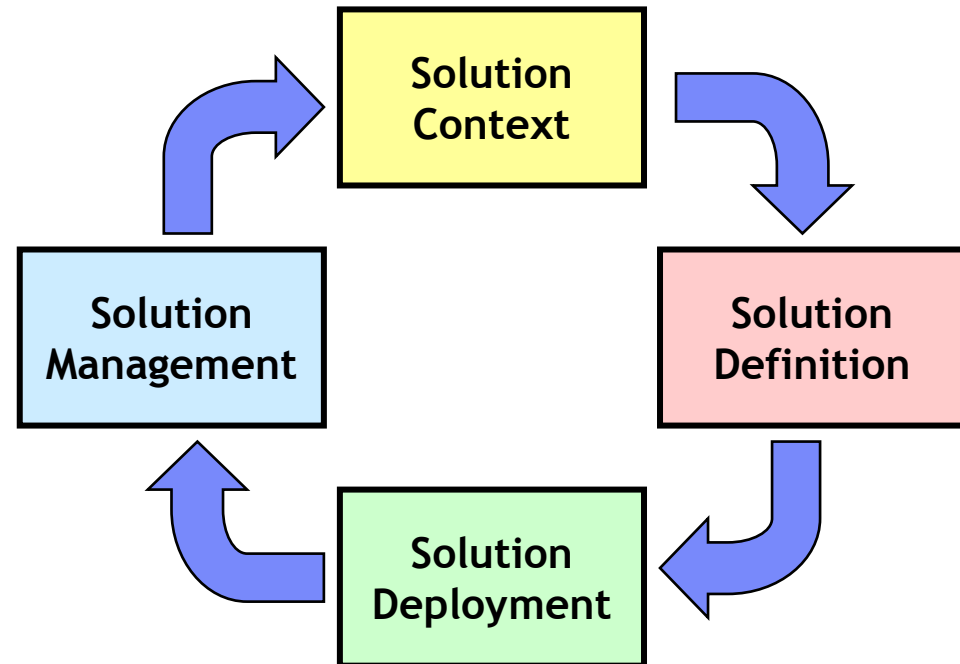


Required Functionality is Realized by All Elements



Key Considerations

- Solution Context
 - ▶ The requirements on the development environment
- Solution Definition
 - ▶ The solution elements that comprise the development environment
- Solution Deployment
 - ▶ Concerns when deploying the environment (e.g. on projects)
- Solution Management
 - ▶ Concerns when supporting the operational environment



Each “wave of change / iteration”, by definition, changes the context for the next wave

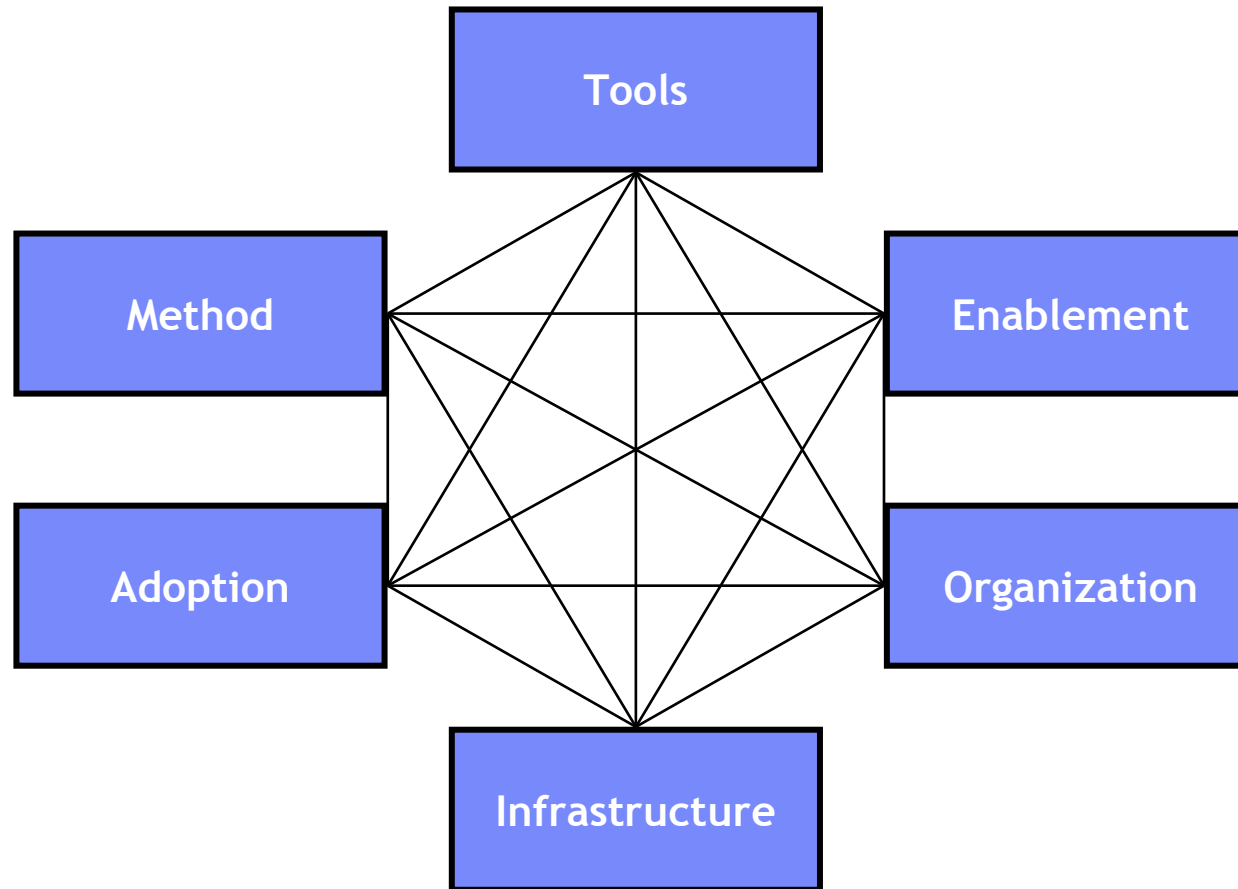
Solution-focused Elements

	Solution Definition	Solution Deployment	Solution Management
Method	Roles, work products, tasks, processes Standards, guidelines, checklists etc. Method deployment topology	Define local configuration Deploy method	Gather feedback on method
Tools	Development tools and integrations Development tool config., install scripts Development tool deployment topology	Perform local configuration Install tools Migrate local data	Backup / archive / restore data Gather feedback on tools
Enablement	Training curriculum and courses Mentoring materials Enablement deployment topology	Perform local configuration Deploy enablement materials Train practitioners	Mentor practitioners Gather feedback on enablement
Organization	Organizational roles and units Organization deployment topology	Define local configuration Reorganize	Gather feedback on organization
Infrastructure	Locations, nodes and connectivity Supporting software (such as OS)	Define local infrastructure Provision locations, nodes, connectivity Provision supporting software	Onboard / retire infra. as required Gather feedback on infrastructure
Adoption	Adoption plan Techniques for driving org. change Environment metrics	Define local adoption plan Validate the environment	Measure environment effectiveness Gather feedback on adoption



Solution Elements are Inter-related

- Many used the defined techniques (method) for making tools (tools) available (adoption) to defined practitioners (organization) by configuring the license server (infrastructure) appropriately



White Paper

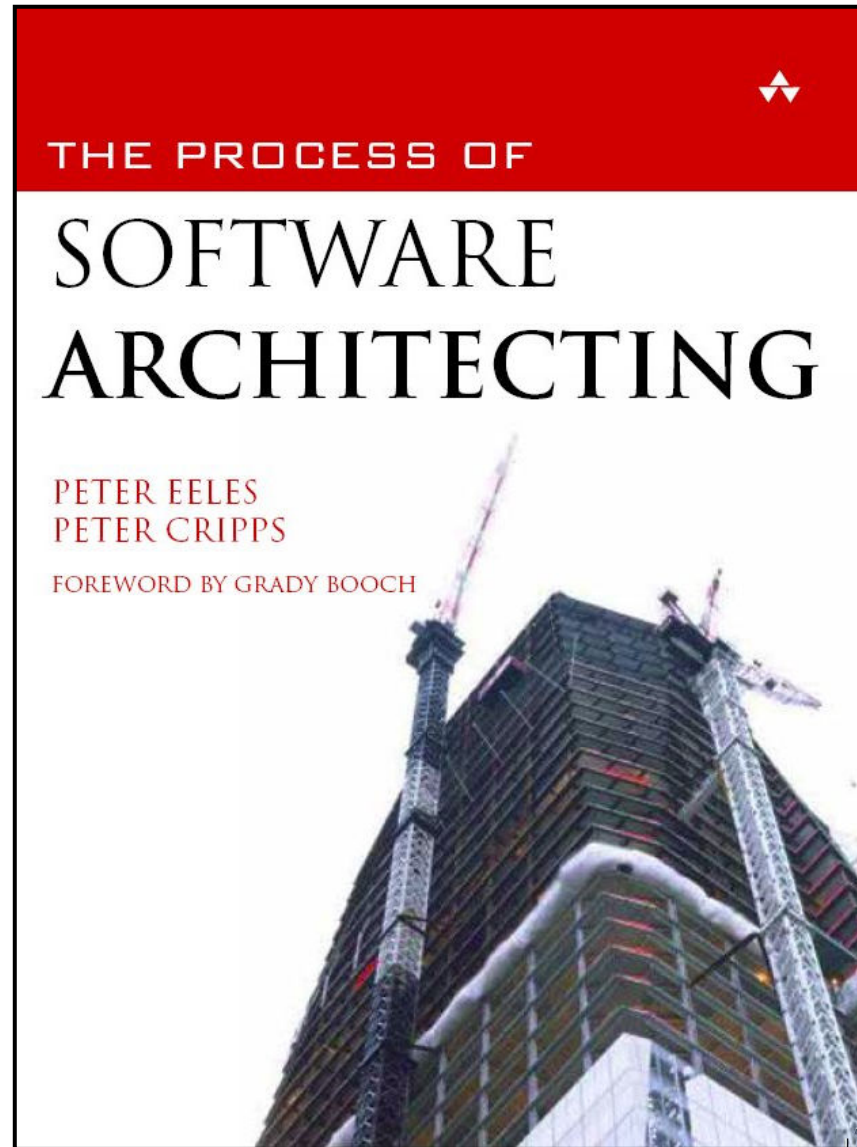


Agenda

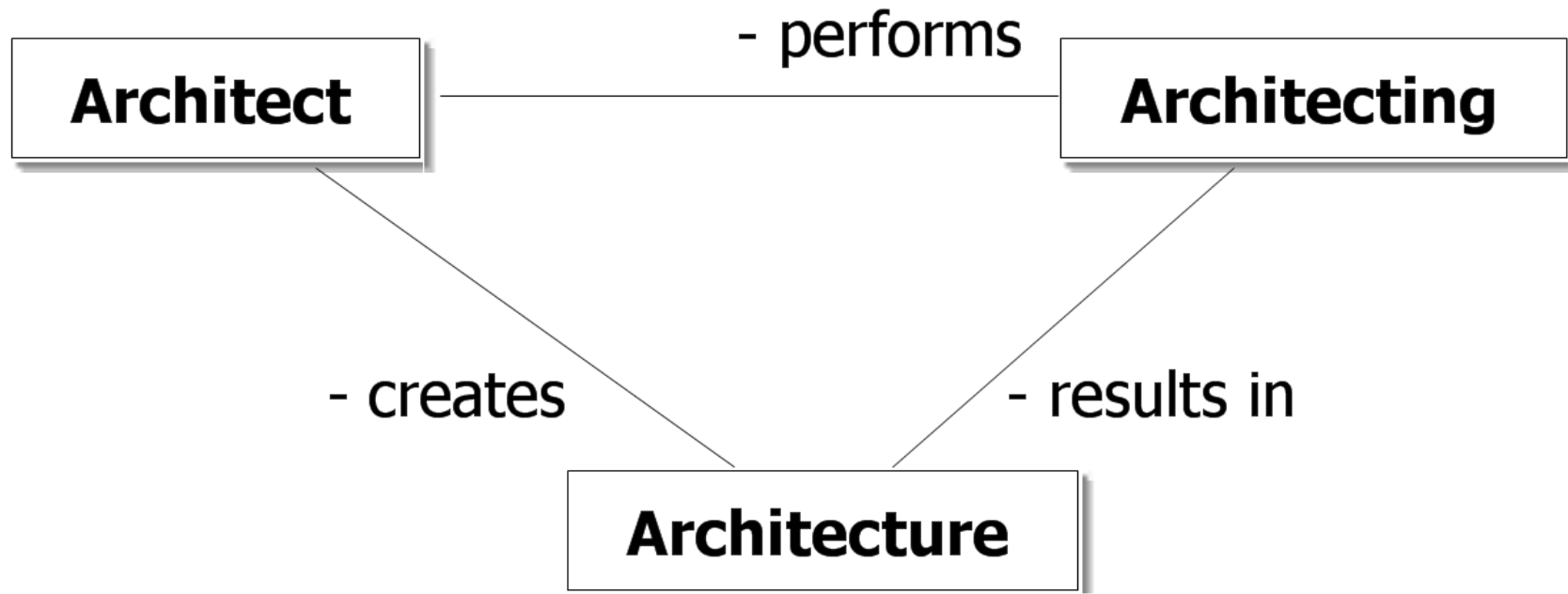
- Start
- Development environment slides
- ➔ Architect slides
 - Development environment architect slides
- Finish



Key Reference



Architecture 101



Software Architecture Defined

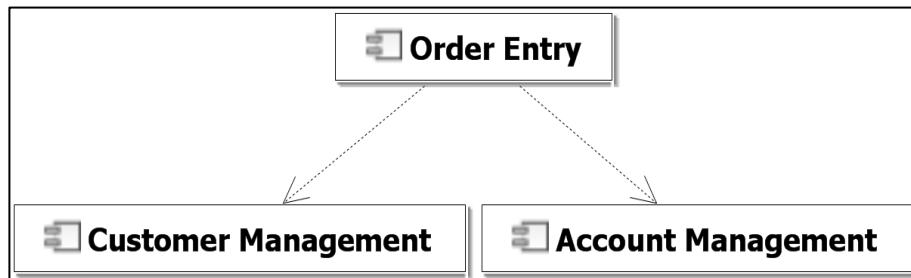
- *Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution. [IEEE 1471]*
- *The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. [Bass 2003]*
- *The software architecture of a system or a collection of systems consists of all the important design decisions about the software structures and the interactions between those structures that comprise the systems. The design decisions support a desired set of qualities that the system should support to be successful. The design decisions provide a conceptual basis for system development, support, and maintenance. [McGovern 2004]*



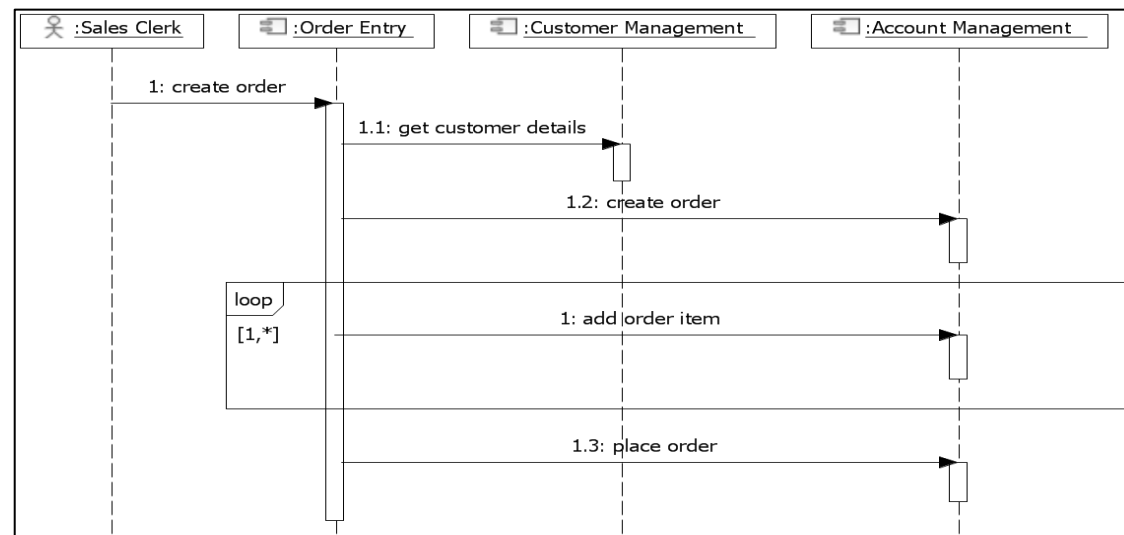
Characteristics of a Software Architecture

- An architecture defines structure
- An architecture defines behaviour

Structure



Behavior



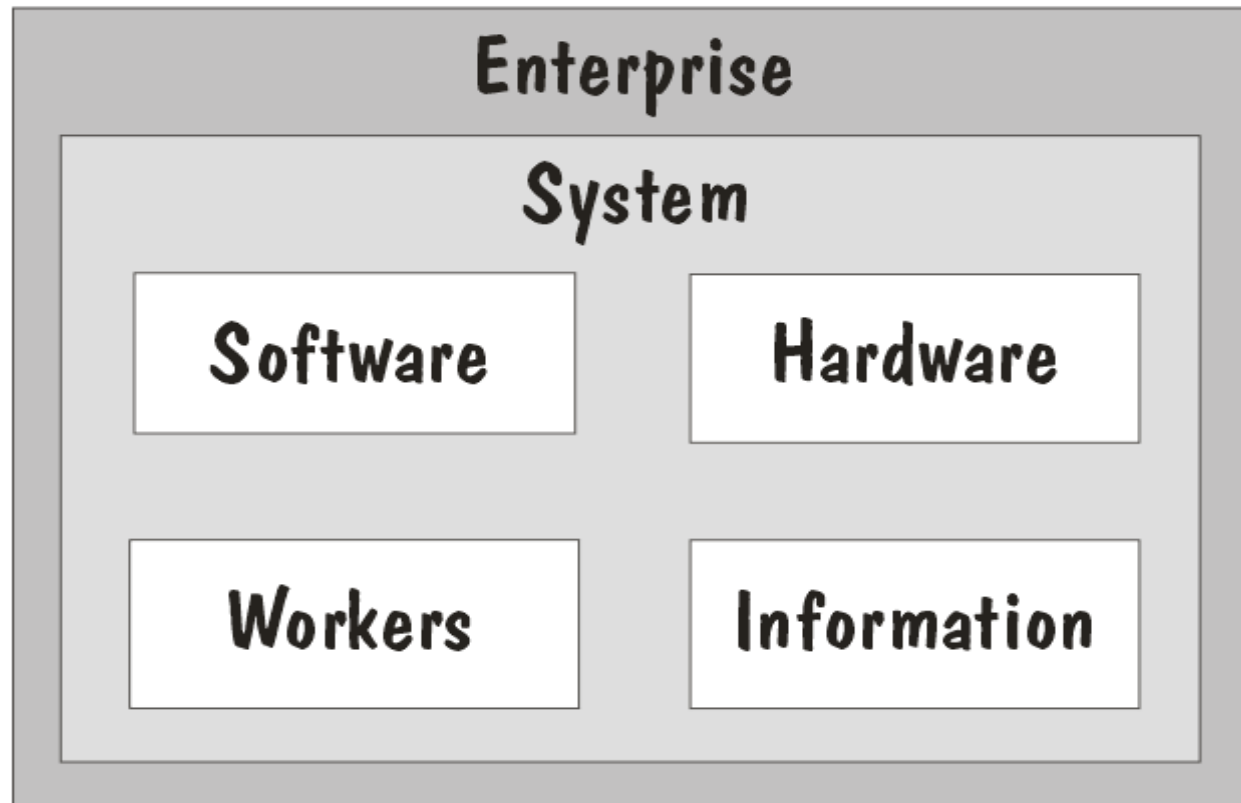
Characteristics of a Software Architecture

- An architecture balances stakeholder needs
- An architecture is influenced by its environment
- An architecture is concerned with significant elements
- An architecture may conform to an architectural style
- An architecture influences development team structure
- An architecture is present in every system
- An architecture embodies decisions based on rationale



Characteristics of a Software Architecture

- An architecture has a particular scope



[A system is] a set of resources that provide services that are used by an enterprise to carry out a business purpose or mission. System components typically consist of hardware, software, data, and workers. [Cantor 2003]



Characteristics of an Architect

- The architect is a technical leader
- The architect understands the development process
- The architect has knowledge of the business domain
- The architect has technology knowledge
- The architect has design skills
- The architect has programming skills
- The architect is a good communicator
- The architect makes decisions
- The architect is aware of organizational politics
- The architect is a negotiator
- The architect role may be fulfilled by a team



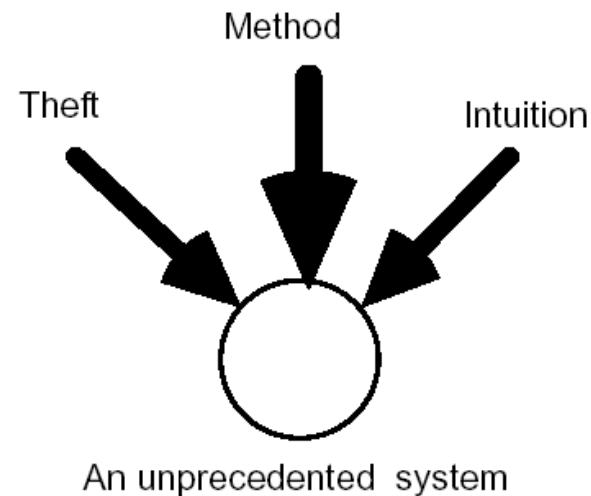
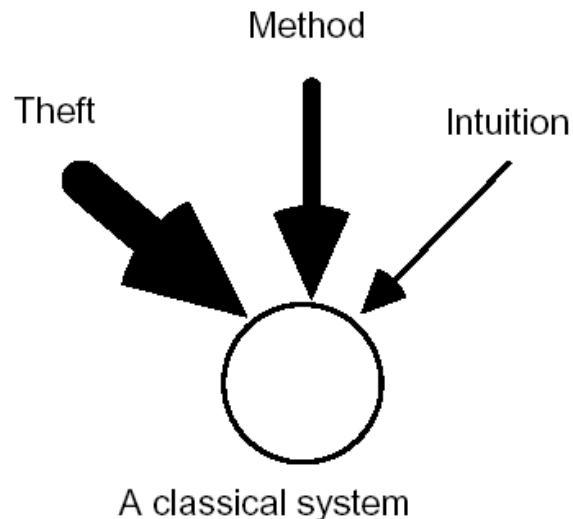
The life of a software architect is a long and rapid succession of suboptimal design decisions taken partly in the dark.

- Philippe Kruchten



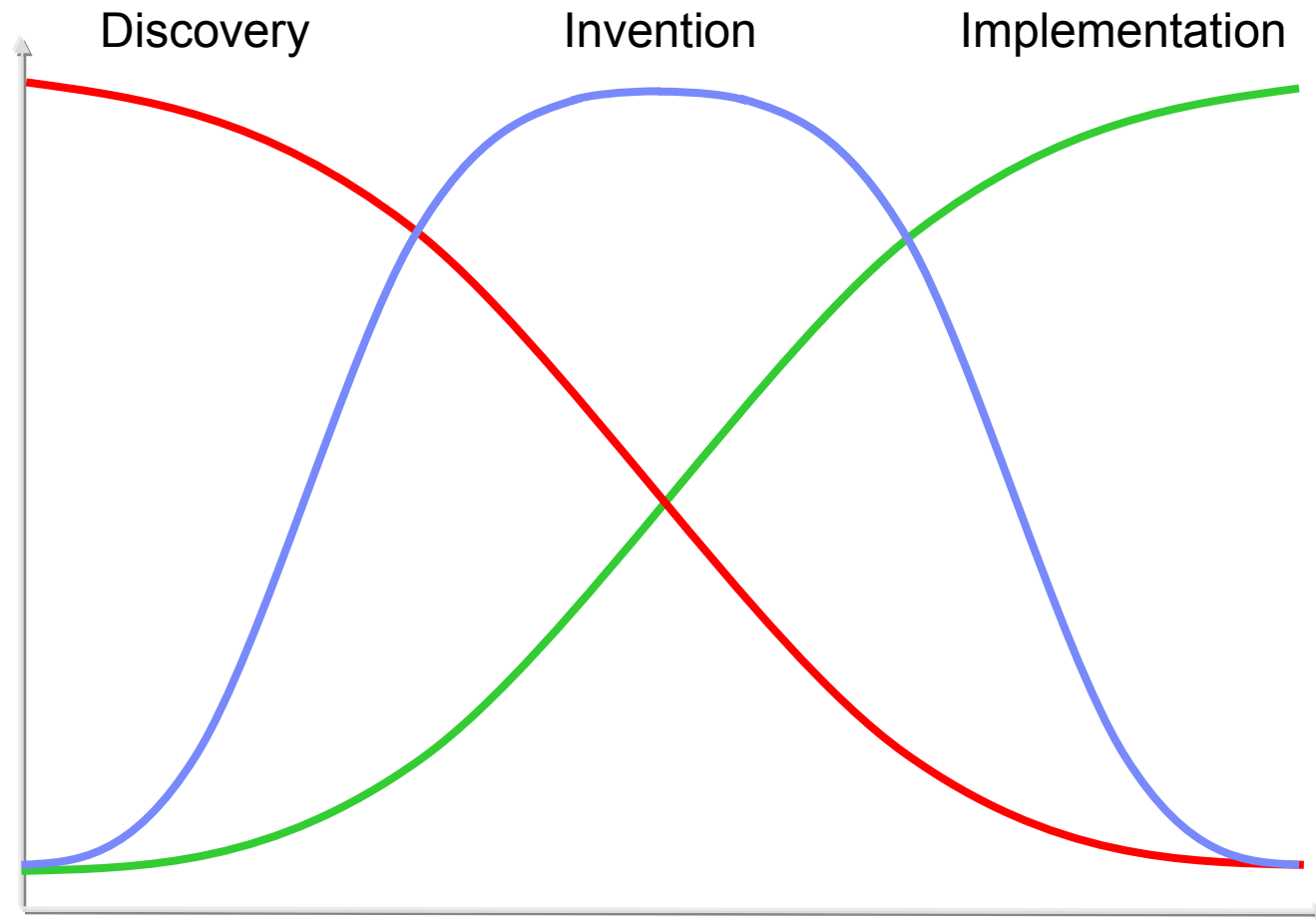
Characteristics of Architecting

- Architecting is a science
 - ▶ We can apply scientific rigor to what we do in terms of, for example, reusable assets, method, etc.
- Architecting is an art
 - ▶ There is still a need for creativity



Characteristics of Architecting

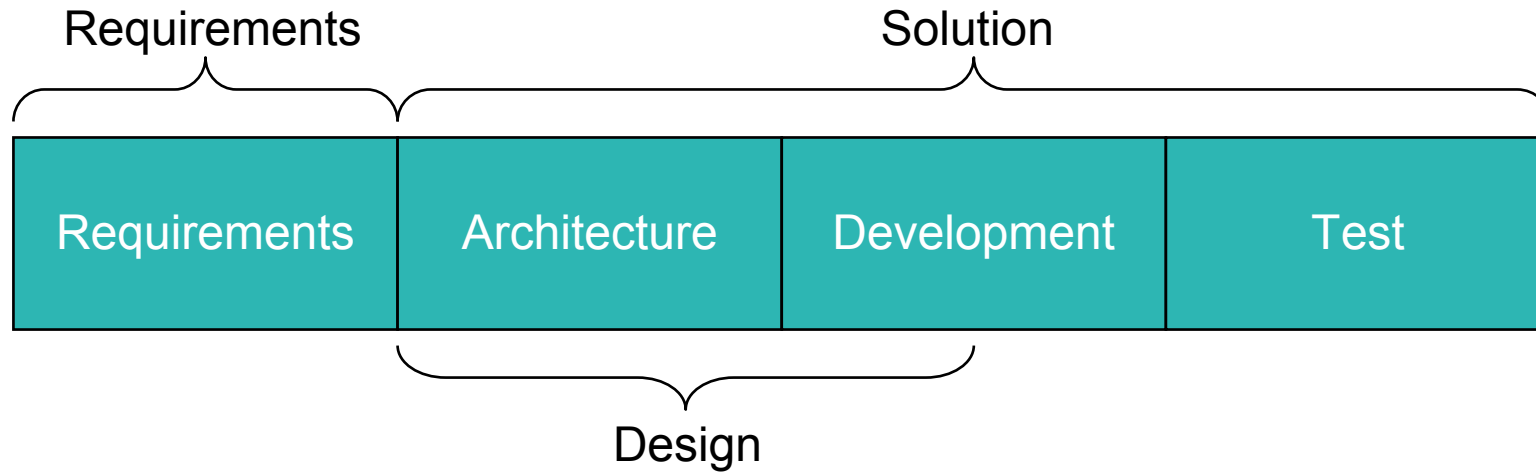
- Architecting changes emphasis over time



Drawn on a napkin by Bran Selic for Philippe Kruchten

Characteristics of Architecting

- Architecting spans many disciplines



All architecture is design but not all design is architecture. Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change.

- Grady Booch

[A discipline is a] primary categorization mechanism for organizing tasks that define a major 'area of concern' and/or cooperation of work effort. [OpenUP 2008]



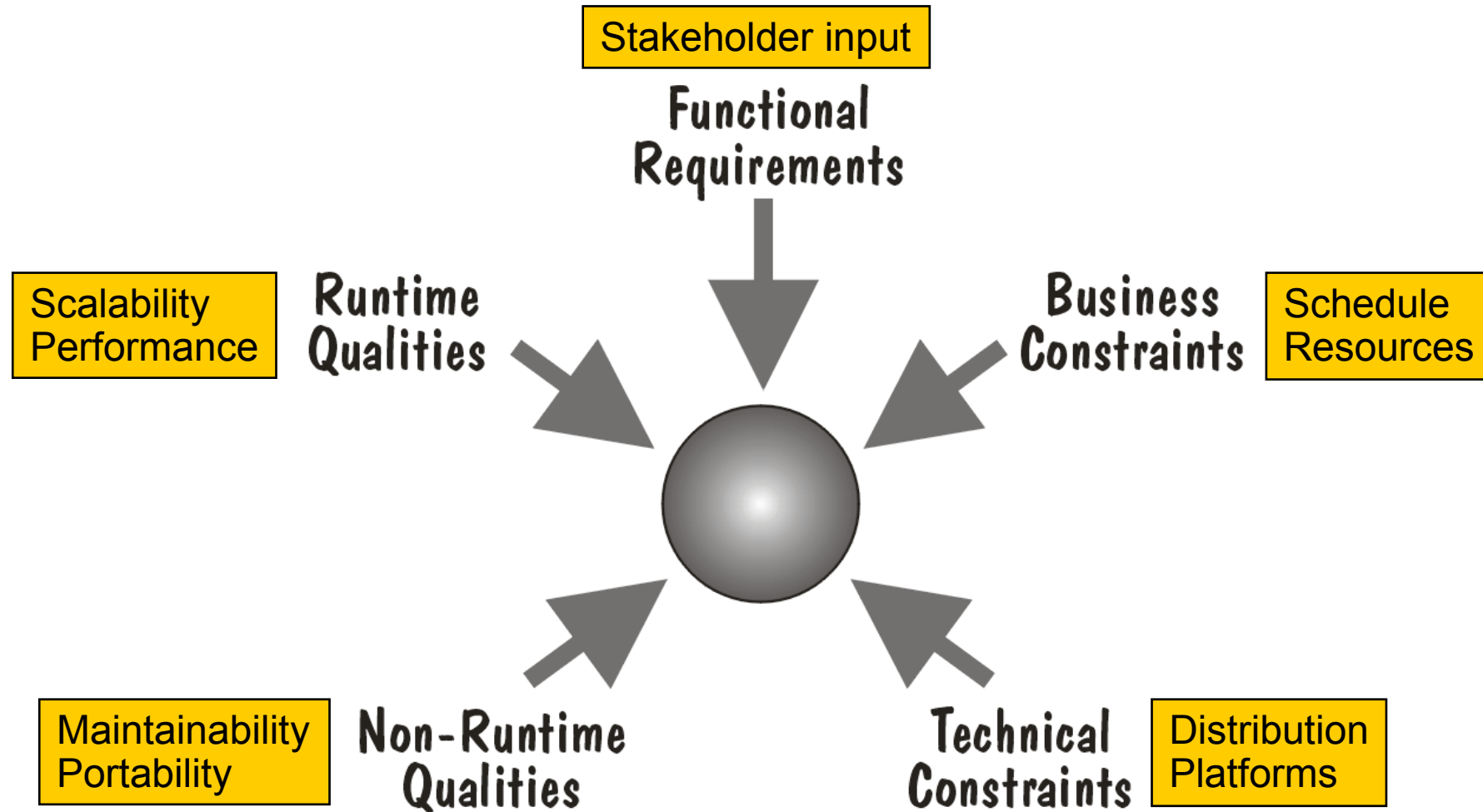
Characteristics of Architecting

- Architecting involves many stakeholders
- Architecting is involved in tradeoffs
- Architecting considers reusable assets
- Architecting is both top-down and bottom-up



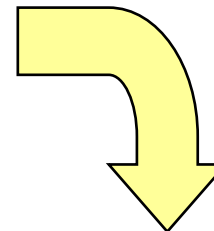
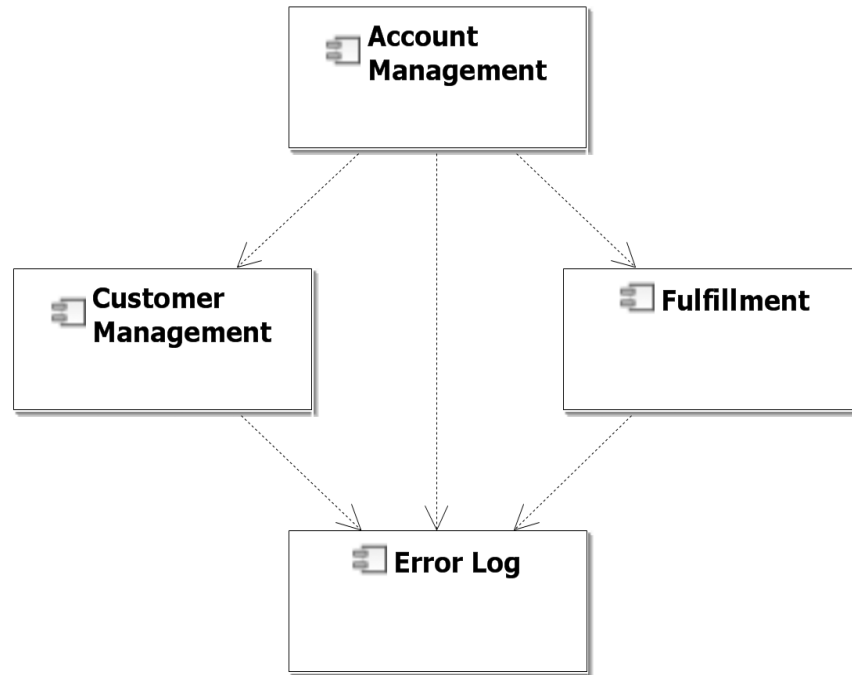
Benefits of Architecting

- Architecting addresses system qualities



Benefits of Architecting

- Architecting supports the planning process



Implement Error Log	[]		
Implement Customer Management		[]	
Implement Fulfillment		[]	
Implement Account Management			[]



Benefits of Architecting

- Architecting helps manage complexity
- Architecting ensures architectural integrity
- Architecting reduces maintenance costs
- Architecting drives consensus
- Architecting provides a basis for reuse
- Architecting supports impact analysis



Agenda

- Start
- Development environment slides
- Architect slides
- ➔ Development environment architect slides
- Finish

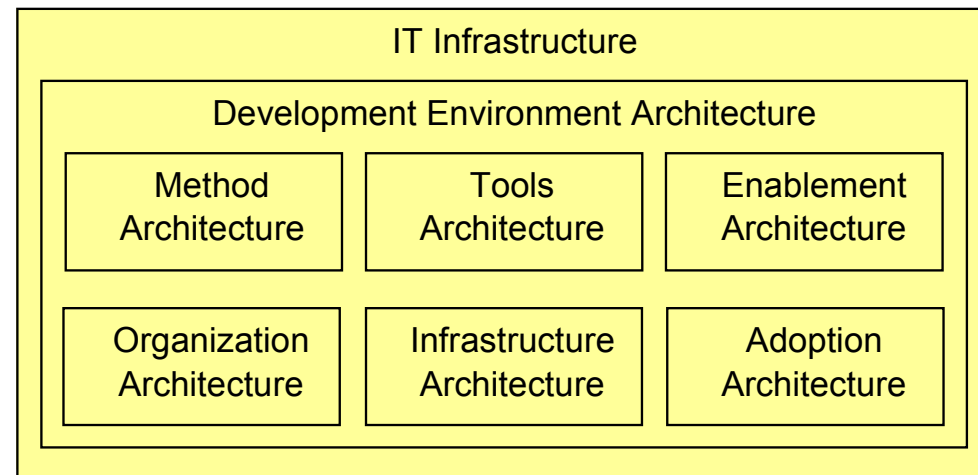
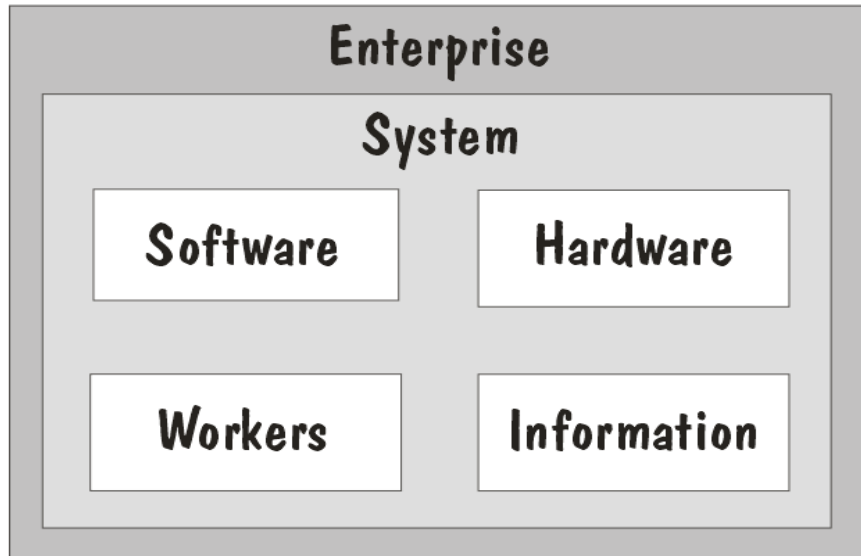


An Architecture is concerned with Significant Elements

- Significant elements
 - ▶ Relate to some critical functionality of the system
 - ▶ Relate to some critical property of the system
 - ▶ Relate to a particular architectural challenge
 - ▶ Are associated with a particular technical risk
 - ▶ Relate to a capability that is considered to be unstable
 - ▶ Relate to some key element of the solution
- Development environment-specifics
 - ▶ Method roles & responsibilities, work products, governance policies
 - ▶ Tools selection, integrations, licensing
 - ▶ Enablement curriculum
 - ▶ Organization roles & responsibilities
 - ▶ Infrastructure distribution, development environment packaging



An Architecture has a Particular Scope



An Architecture meets Stakeholder Needs

- Typical stakeholders of a development environment



- ▶ Practitioner

- Intuitive and correct behavior, performance, reliability, usability, availability, security

- ▶ System administrator

- Intuitive behavior, administration, tools to aid monitoring

- ▶ Customer

- Cost, return on investment, stability, schedule



- ▶ Implementers

- Clear requirements, simple and consistent design approach

- ▶ Maintainer

- Comprehensible, consistent and documented design approach, ease with which modifications can be made



- ▶ Sponsor

- Alignment of anticipated results with business and IT strategy

- ▶ Strategic suppliers

- Providing tools, training, infrastructure and second or third line support



An Architecture is Influenced by its Environment

- Method Regulatory / organizational standards
- Tools Existing “standard” tooling
- Training An existing training curriculum
- Organization Existing skills, organizational structures
- Infrastructure Existing infrastructure
- Adoption Approach to on-boarding teams on projects



Architecting addresses System Qualities

- Method Usability, configurability, ...
- Tools Performance, scalability, ...
- Training Usability, ...
- Organization Evolvability, ...
- Infrastructure Performance, ...
- Adoption Return on investment, ...



Agenda

- Start
- Development environment slides
- Architect slides
- Development environment architect slides

 Finish



Summary

- Development Environment Architects are ... Architects!
- Through analogy with a software architect, we can better-understand the role of the Development Environment Architect
 - ▶ Architecture
 - ▶ Architect
 - ▶ Architecting
 - ▶ Benefits of architecting
- This is an emerging domain that is becoming increasingly important!



THANK YOU