Software – Role in Systems and their Architectures

Abhilash Gopalakrishnan

Teachers

"'India with her mighty scientific knowledge & power house of young should build her own huge rocket systems and also build her own communication, remote sensing and meteorological spacecraft and launch from her own soil to enrich the Indian life in satellite communication, remote sensing & meteorology. The projects selected in space programme, are designed to meet the societal needs" –Prof Vikarm Sarabhai in 1970.

We witnessed it transform to truth!

'A Promise is a Promise' – Ratan Tata

'I believe there is no other profession in the world that is more important to the society than that to a teacher'- Dr. APJ Abdul Kalam

Indomitable Spirit

I take this opportunity to thank all my teachers - school, college or workplace!

Great Minds at Work!

Agenda

- Alignment to the Program
- Systems and Increasing Software Intensity
- IEEE 1471 standard
- Architecture Business Cycle
- Problem Space and Solutions space
- Architecture Definition

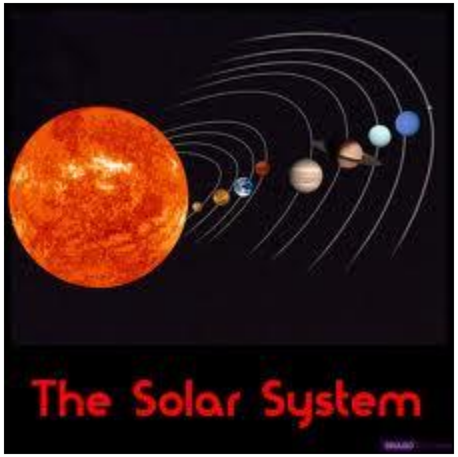
Technology Leadership and Architecture

Courtesy: Facebook.com

- ❑ Evolution of Computer Programming Languages indicate increasing levels of abstraction.
- ❑ Significance moves to structures and organize them :- Architecture
- ❑ Architecture plays a key role in Innovations

Technology Leadership is key to sustainability

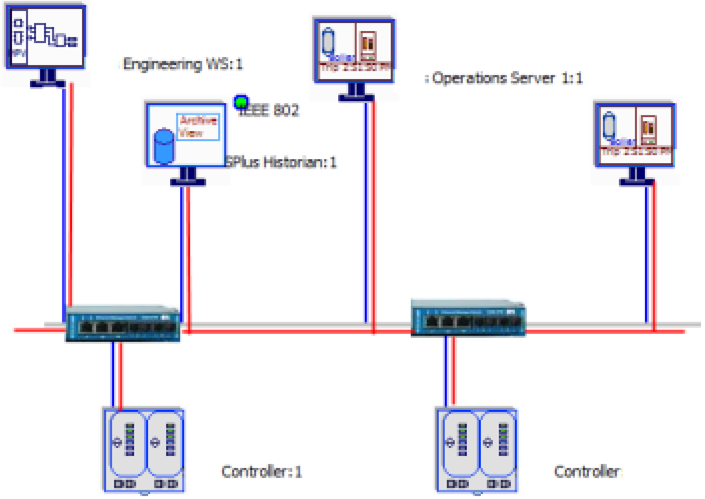
Systems



- Systems form a chain involving
  - mechanical, electrical, electronics, communication & software entities.
  - Involves multiple interfaces across various subsystems
  - The lifecycle of these systems normally longer.
  - Range from embedded systems for control or automotive applications, telecommunication, business applications like banking.

The complexities and size aspects of systems might vary.

For someone from ISRO it is Earth as a System,
For someone from automobile industry it is about vehicular systems like Anti Braking System (ABS)
For someone from Industrial Control Systems it is SCADA or DCS.



Distributed Control System

We limit our discussion to more commercial applications like industrial control systems.

Systems span multiple disciplines

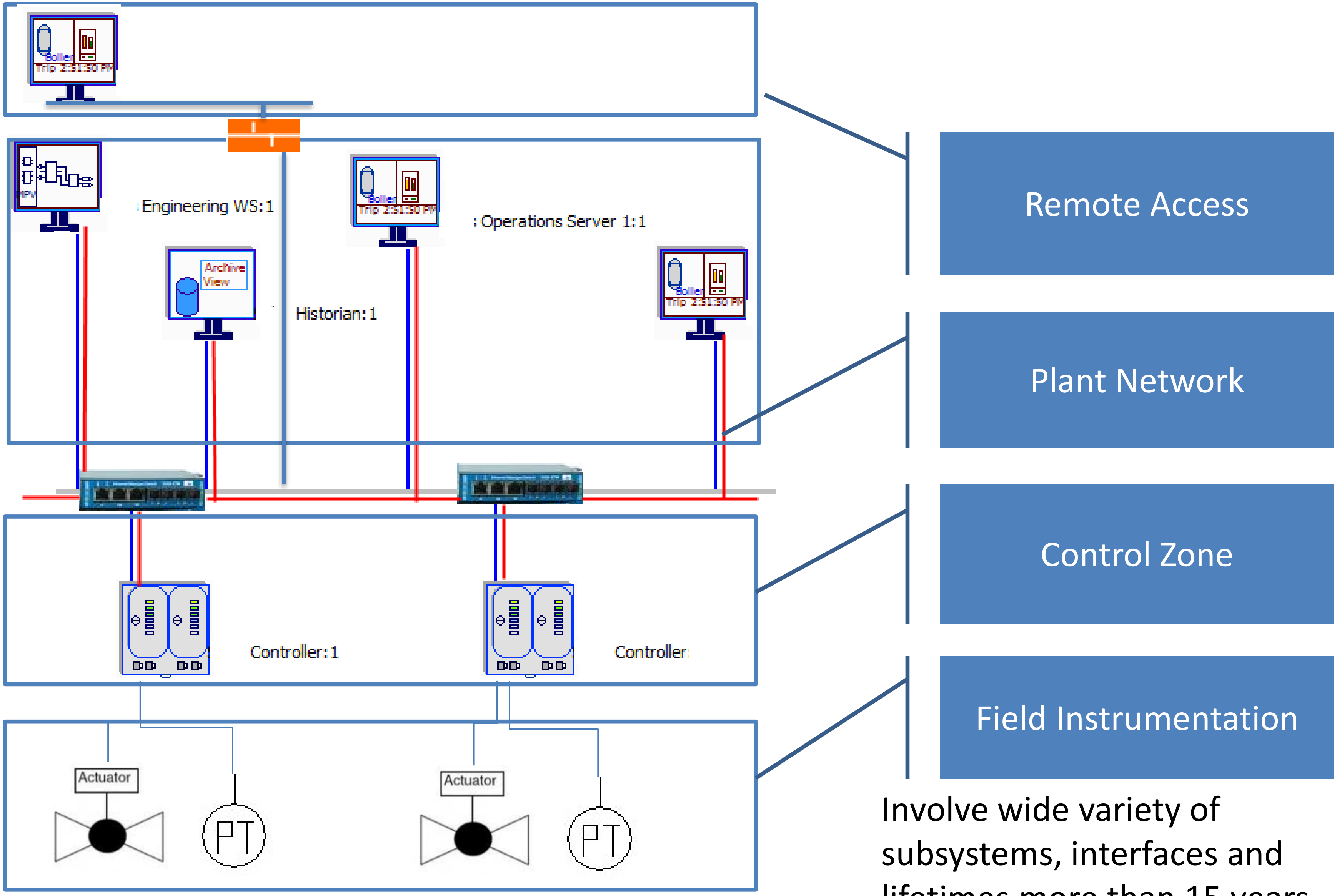
Example : Asian Paints Distribution Network

Paint dispensing machines integrated with Software based HMI found at Asian Paint dealerships

Courtesy: Asian Paints Dealership

Systems have become an invisible part of our lives!

Example: Industrial Control systems



Remote Access

Plant Network

Control Zone

Field Instrumentation

Involve wide variety of subsystems, interfaces and lifetimes more than 15 years

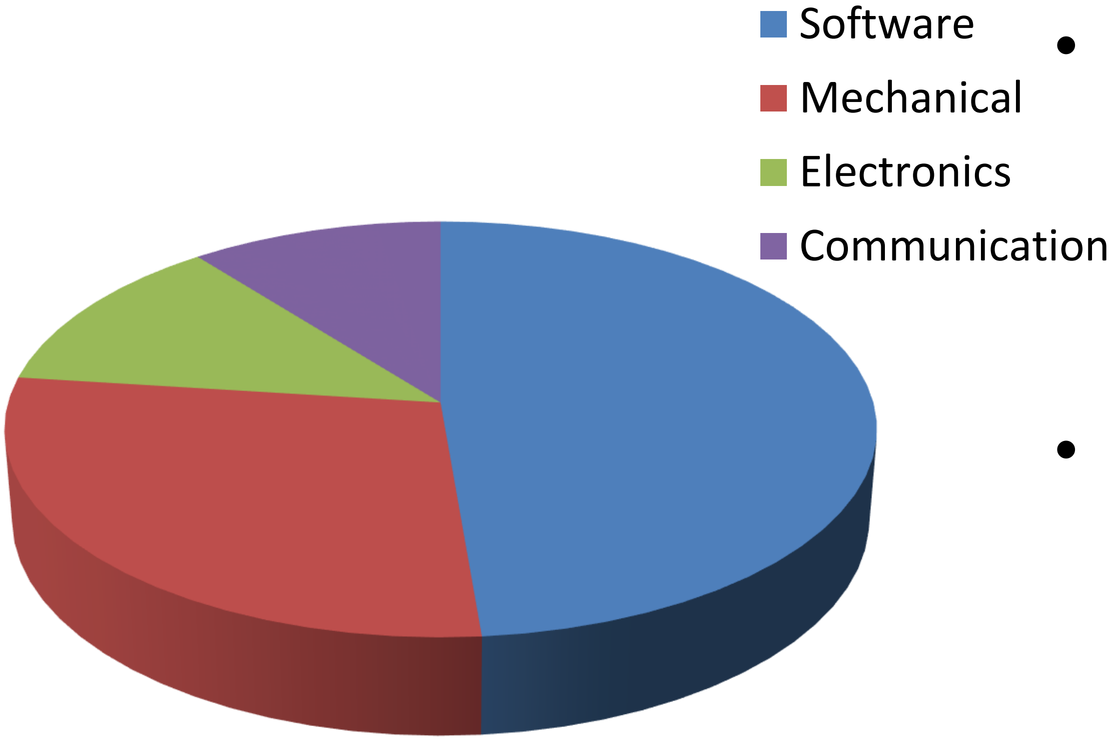
Systems and IT Applications

Characteristic	Systems	IT Applications
Time to market	1 Year and reducing	Much less, even in the range of months
Target Market	Specific market segments driven by reliability and cost factors	Targeted at applications more to improve productivity.
Expected Lifetime	10-30 years	1-5 years
Relevant owners	Phase wise owners from operator, engineer, end customer, decommissioning engineer	Owners could vary across projects.
Legacy support/Backward Compatibility Requirements	Strong requirements due to huge investments required	Weak requirements due to the fact that an IT implementation could reduce current operating cost.
Shutdown permitted	Shutdown only permitted as part of planned maintenance. Else could lead to losses	Shutdown is permitted within tolerance benchmarks.
Safety Critical	Yes, considering human lives and huge investments to be protected	Few implementations are safety critical, but not necessarily

Agenda

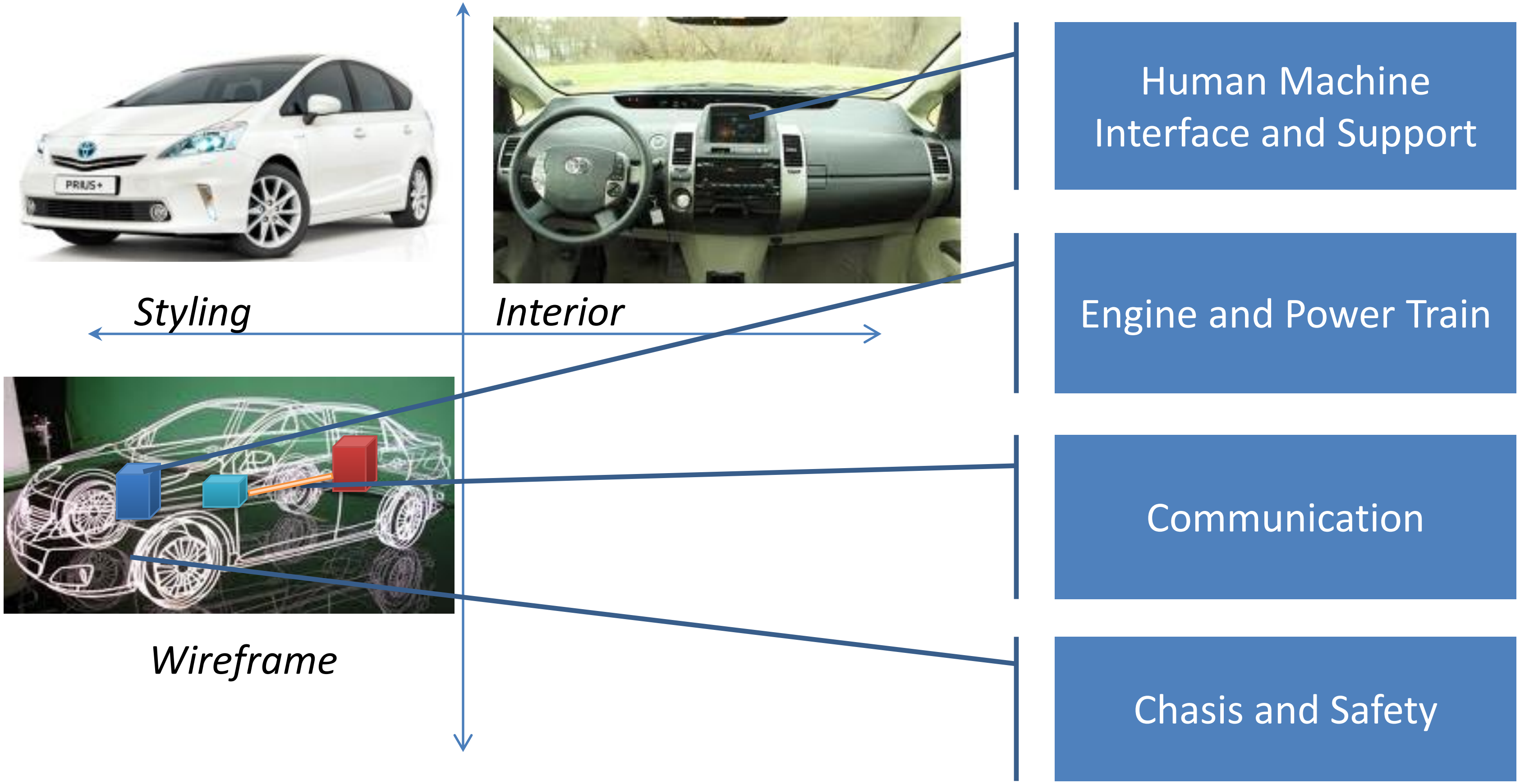
- Alignment to the Program
- Systems and Increasing Software Intensity
- IEEE 1471 standard
- Architecture Business Cycle
- Problem Space and Solutions space
- Architecture Definition

Software Intensive Systems



- Systems can be classified software intensive when,
  - 40-50 % of subsystems constitute software
- Examples
  - Vehicular Systems it could even be 50% Software components
  - Industrial Control Systems is close to 40%

Vehicular Systems and Software Intensive areas



Styling

Interior

Wireframe

Human Machine Interface and Support

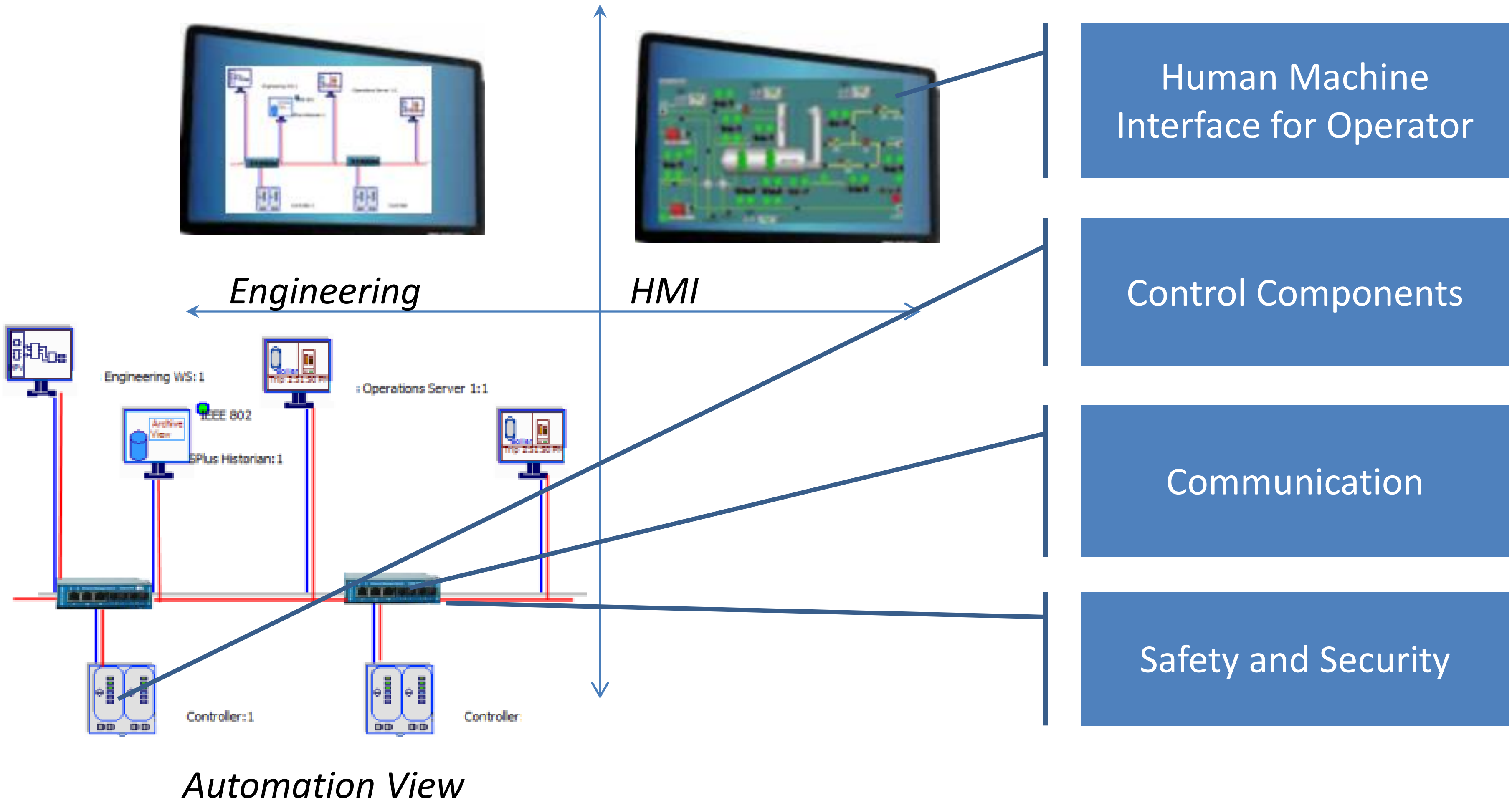
Engine and Power Train

Communication

Chasis and Safety

Software is a key ingredient

Industrial Control Systems and Software Intensive areas



Human Machine Interface for Operator

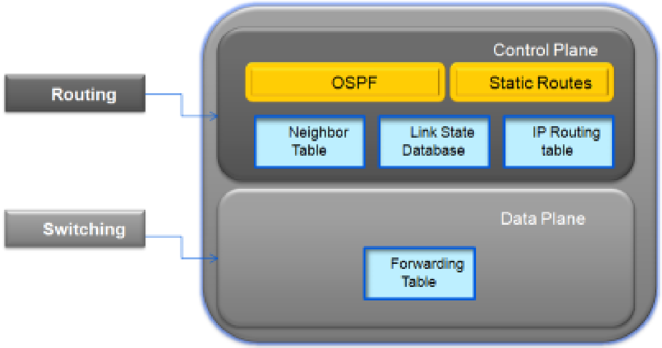
Control Components

Communication

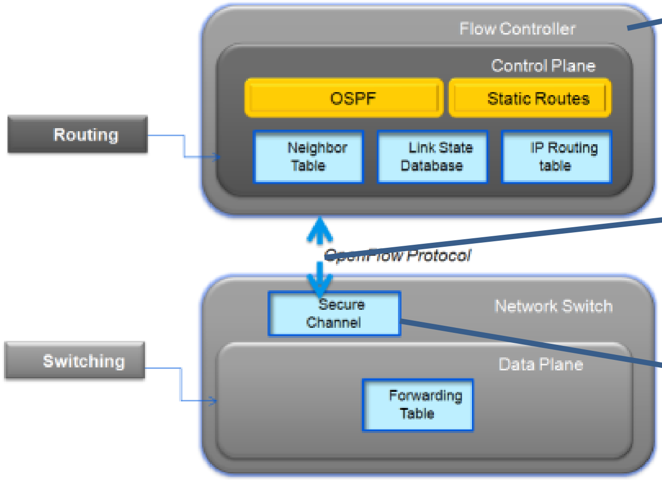
Safety and Security

Software is emerging as a key player

Telecommunication: growing intensity of software



Existing Switch Architectures



Software Defined Network based Architecture

Hardware and Software intermingled/ Closed Architecture

Flow Controller Software would handle flow

New Protocol introduced

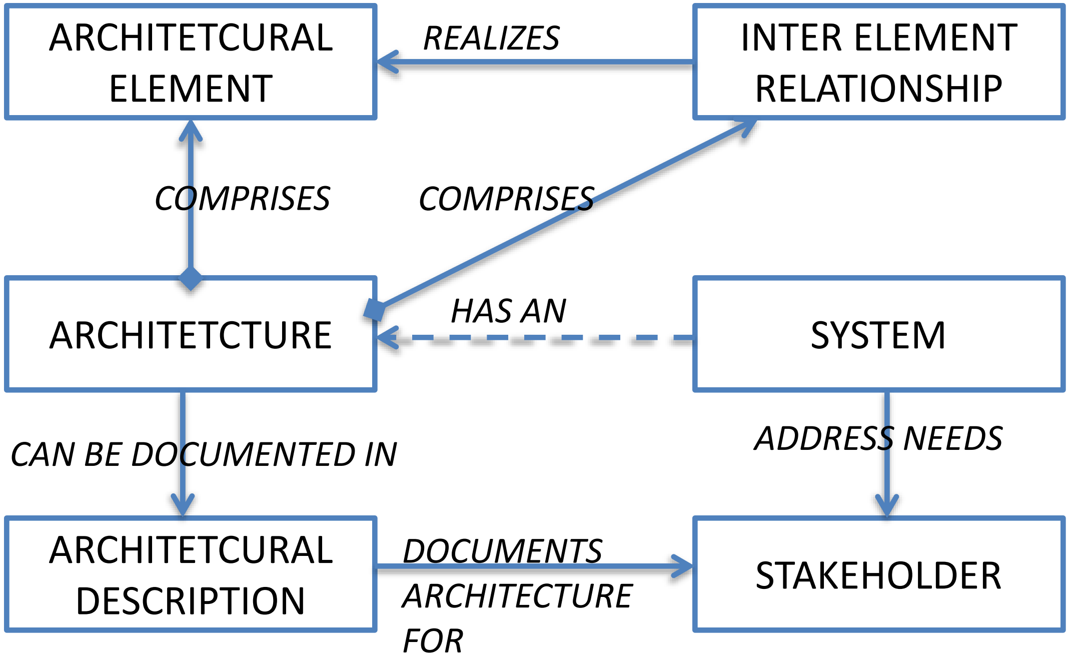
Safety and Security

Software plays a key role in innovation

Agenda

- Alignment to the Program
- Systems and Increasing Software Intensity
- IEEE 1471 standard
- Architecture Business Cycle
- Problem Space and Solutions space
- Architecture Definition

IEEE 1471 – On Software Architecture



- A system is built to address the needs concerns goals and objectives of its stakeholders (owners)
- The architecture of system comprises of its architectural elements and their interrelationships
- Architecture can be documented in Architecture Description
- An Architecture Description documents architecture for the stakeholders and demonstrates to them that it has met their needs

Architecture address needs of stakeholders

Architecture- Quality Attributes

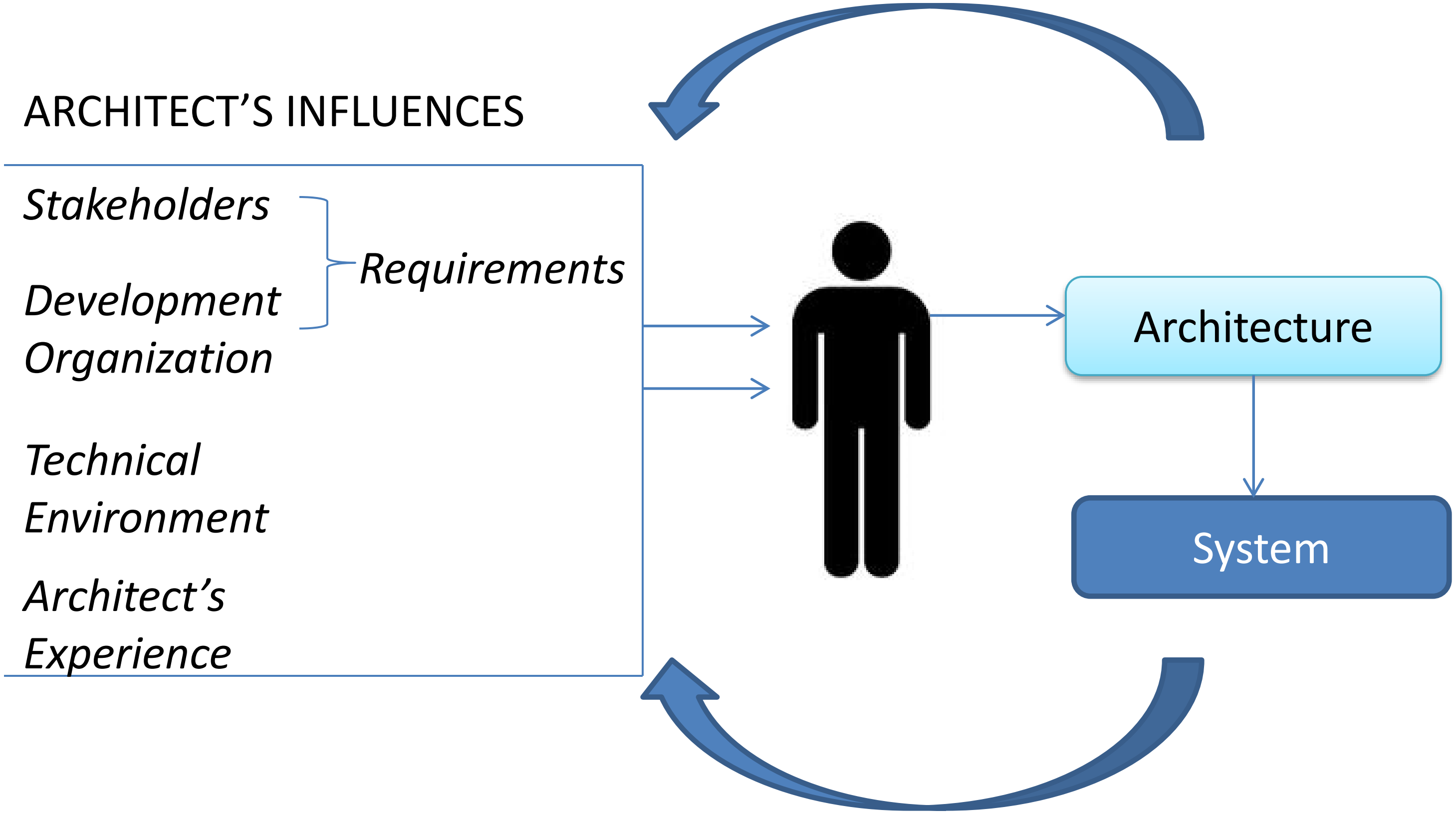
Attribute Name	Industrial Control System	Vehicular Systems
Reliability	High Reliability – Probably of failure is 10^{-9}	High Reliability – Probably of failure is 10^{-9}
Safety	Safety important and applies more to certain types of Process - IEC 61508 SIL /*ANSI*/ISA-84.01-199	ISO 26262 Safety Integrity level for Automobiles
Availability	High Availability and Only Planned Shutdown. Redundancy used	Redundancy used both as node as well as communication paths. Based on application
Security	High Security needs based on zones	Security capabilities exist. Used or not used based on application
Performance	Deterministic Response expected. Hard Real time in control and protection. Ranges between 1 micro second to 1 millisecond precision and 3ms- 20 ms time to respond.	Determinism expected. Differs between application For example Power Train and Chasis are hard real time and Telematic HMI soft real time

Reliability & Safety are at most importance

Agenda

- Alignment to the Program
- Systems and Increasing Software Intensity
- IEEE 1471 standard
- Architecture Business Cycle
- Problem Space and Solutions space
- Architecture Definition

Architecture Business Cycle



ARCHITECT'S INFLUENCES

Stakeholders

Development Organization

Requirements

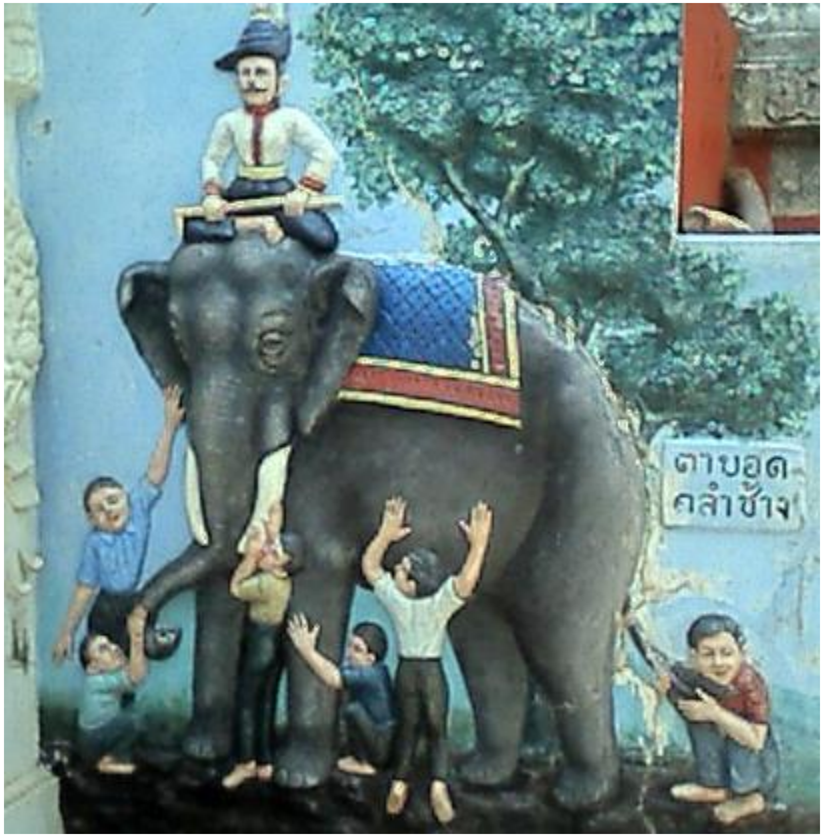
Technical Environment

Architect's Experience

Architecture

System

ViewPoints and Views

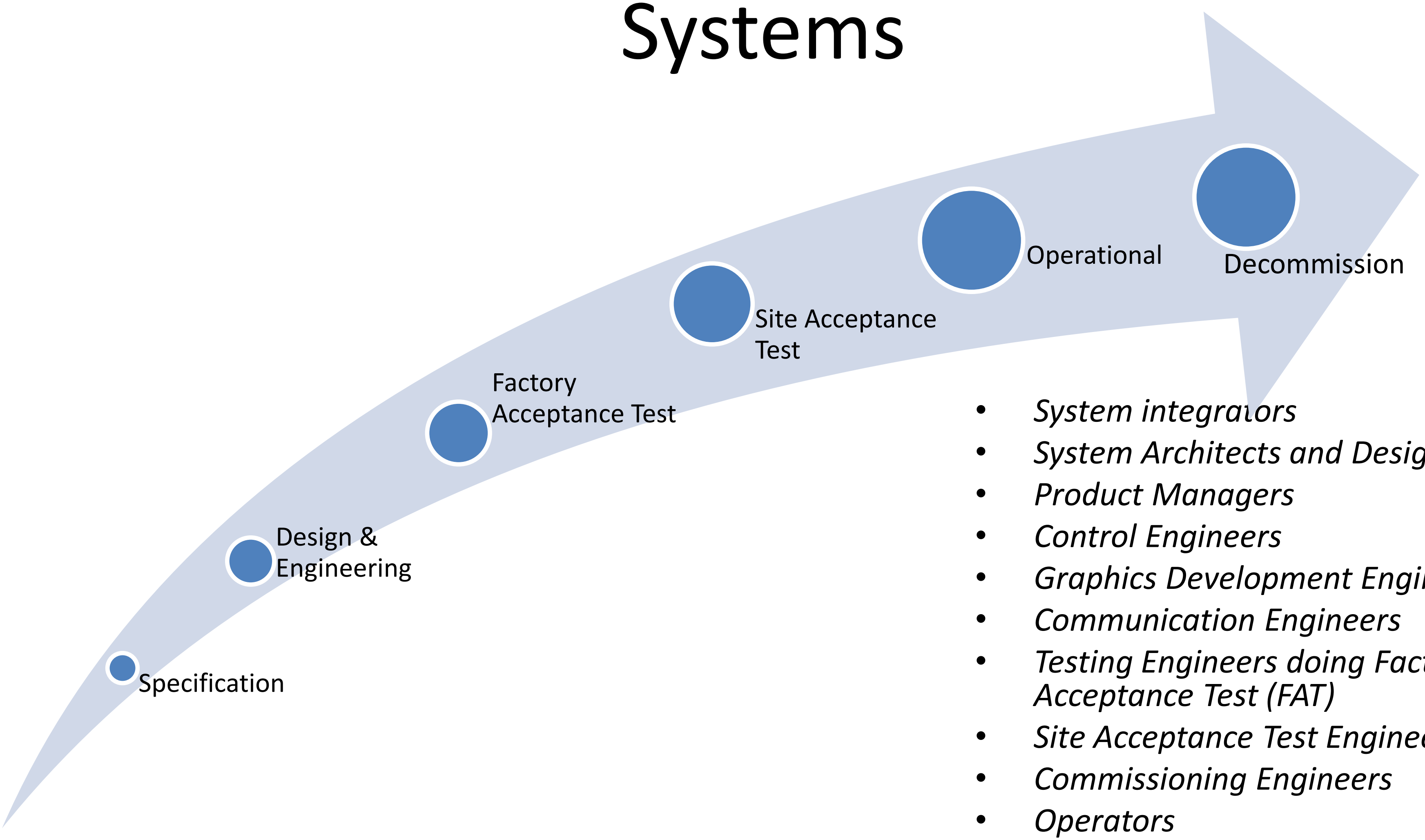


Courtesy: Wikipedia – Elephant and blind men

"*We have to remember that what we observe is not nature in itself, but nature exposed to our method of questioning*.“ -Werner Heisenberg

Architect can see the elephant as a whole

Stakeholders in Industrial Control Systems



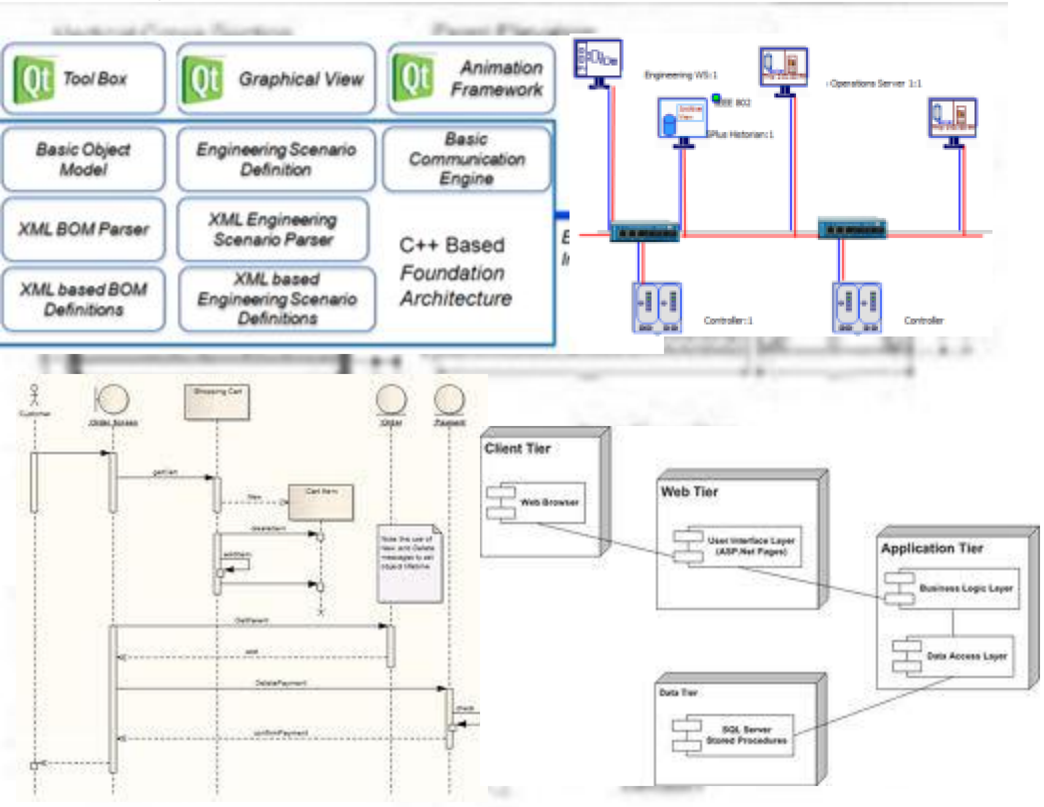
- *System integrators*
- *System Architects and Designers*
- *Product Managers*
- *Control Engineers*
- *Graphics Development Engineers*
- *Communication Engineers*
- *Testing Engineers doing Factory Acceptance Test (FAT)*
- *Site Acceptance Test Engineers*
- *Commissioning Engineers*
- *Operators*
- *Decommissioning Engineers*

It is a challenge to even identify all needs

Significance of Architecture

Development View

Runtime View



Implementation View

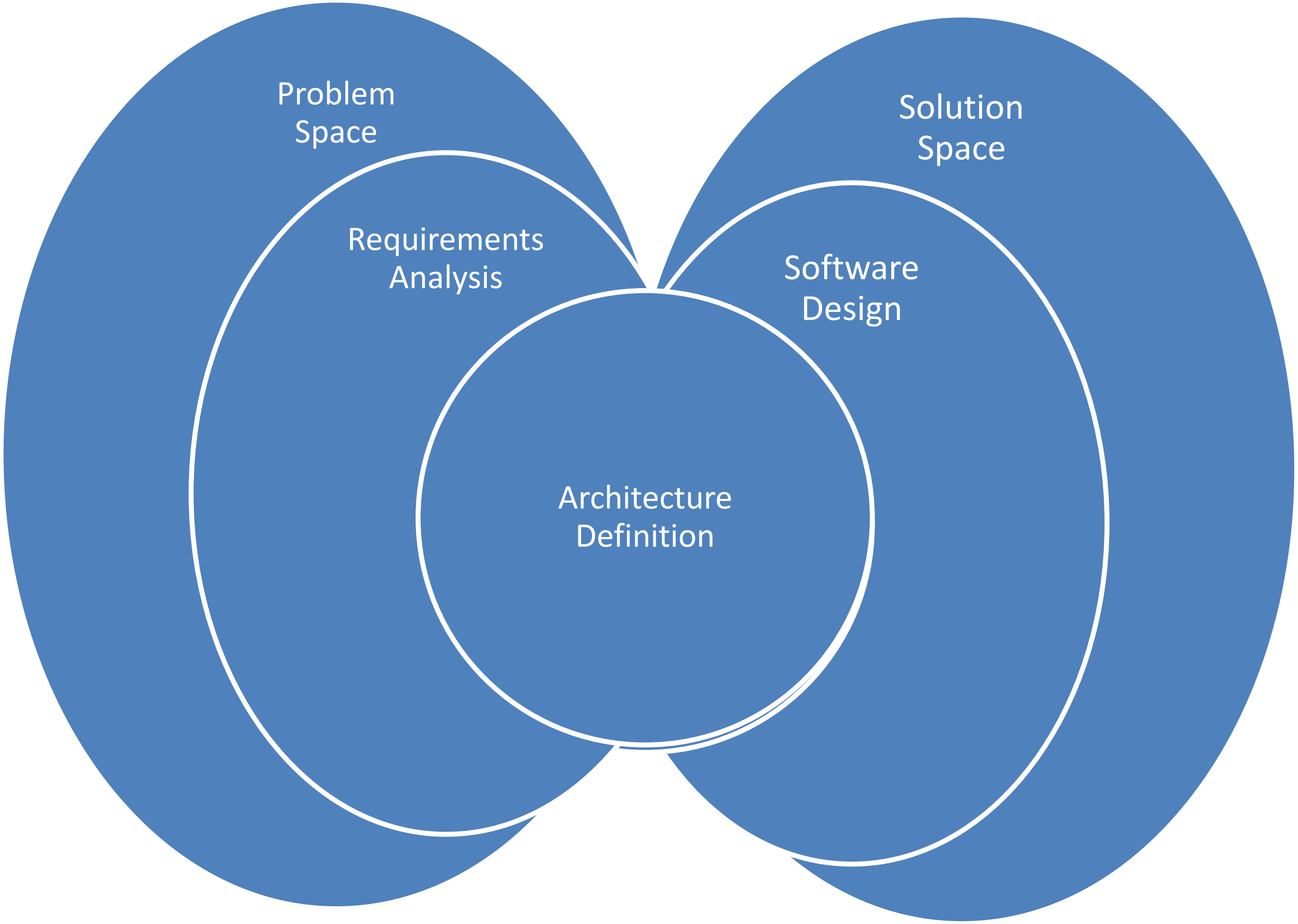
Deployment View

- As a vehicle for communication among stakeholders
- Manifestation of earliest design decisions about a system
- Transferable, reusable abstraction of a system

Agenda

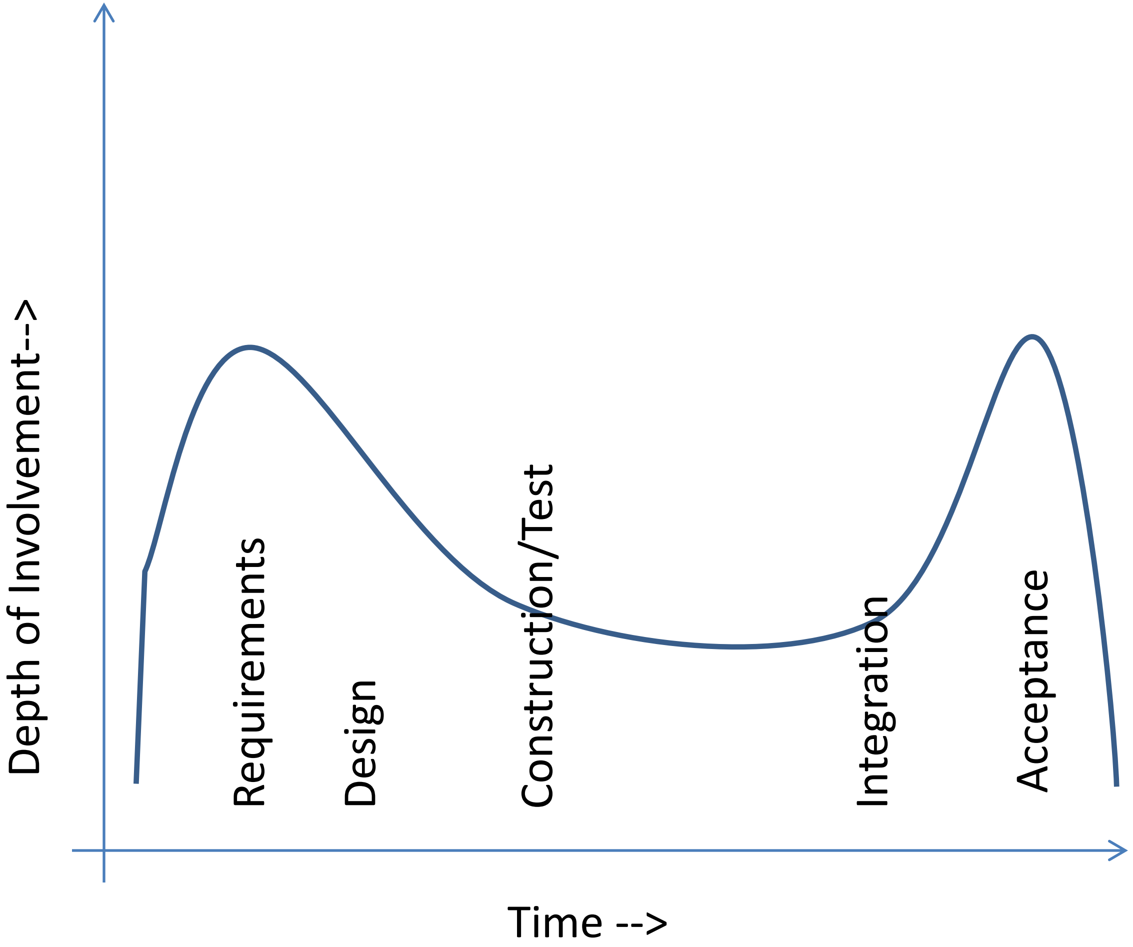
- Alignment to the Program
- Systems and Increasing Software Intensity
- IEEE 1471 standard
- Architecture Business Cycle
- Problem Space and Solutions space
- Architecture Definition

The Pivotal role of Architect



Architect is a stakeholder’s person!

Architect’s Involvement in Development



Depth of involvement of architect could vary across phases in the life cycle.

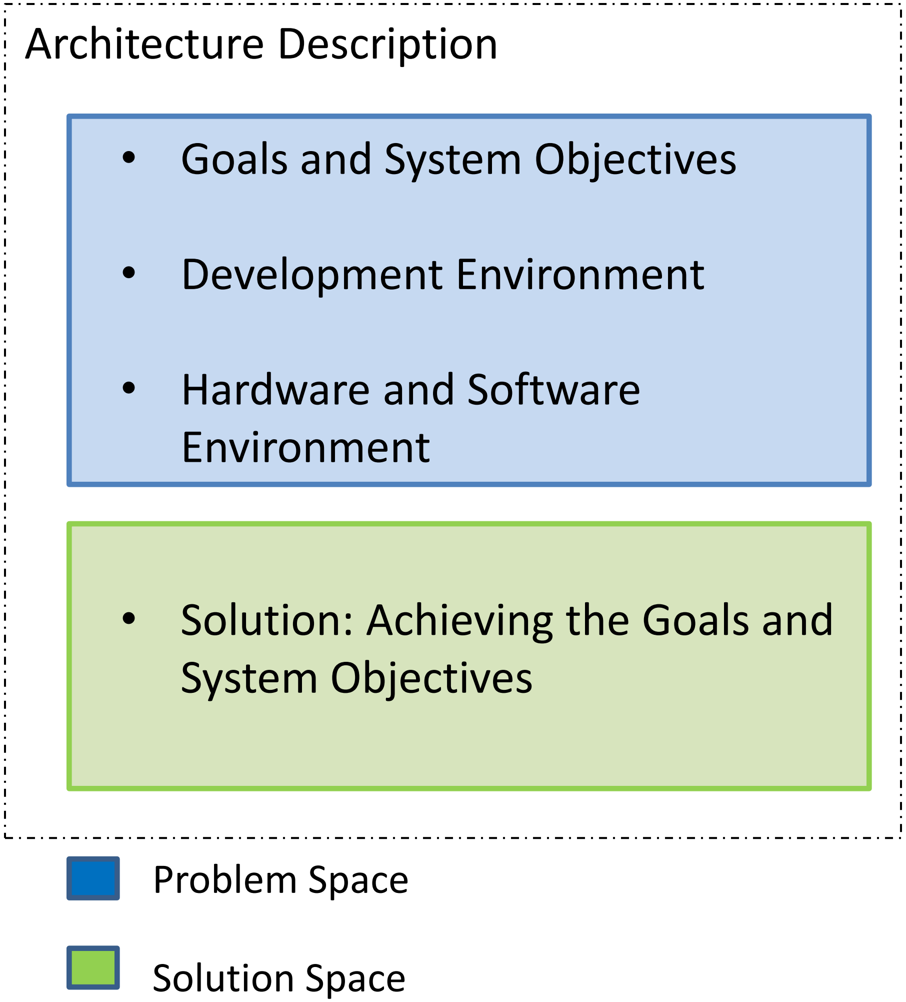
Architect uses his programming skills to build demonstrators showing how requirements are met

Architect is a developer’s person!

Agenda

- Alignment to the Program
- Systems and Increasing Software Intensity
- IEEE 1471 standard
- Architecture Business Cycle
- Problem Space and Solutions space
- Architecture Definition

Architecture Definition



Architecture Description

- Goals and System Objectives
- Development Environment
- Hardware and Software Environment

- Solution: Achieving the Goals and System Objectives

🟦 Problem Space

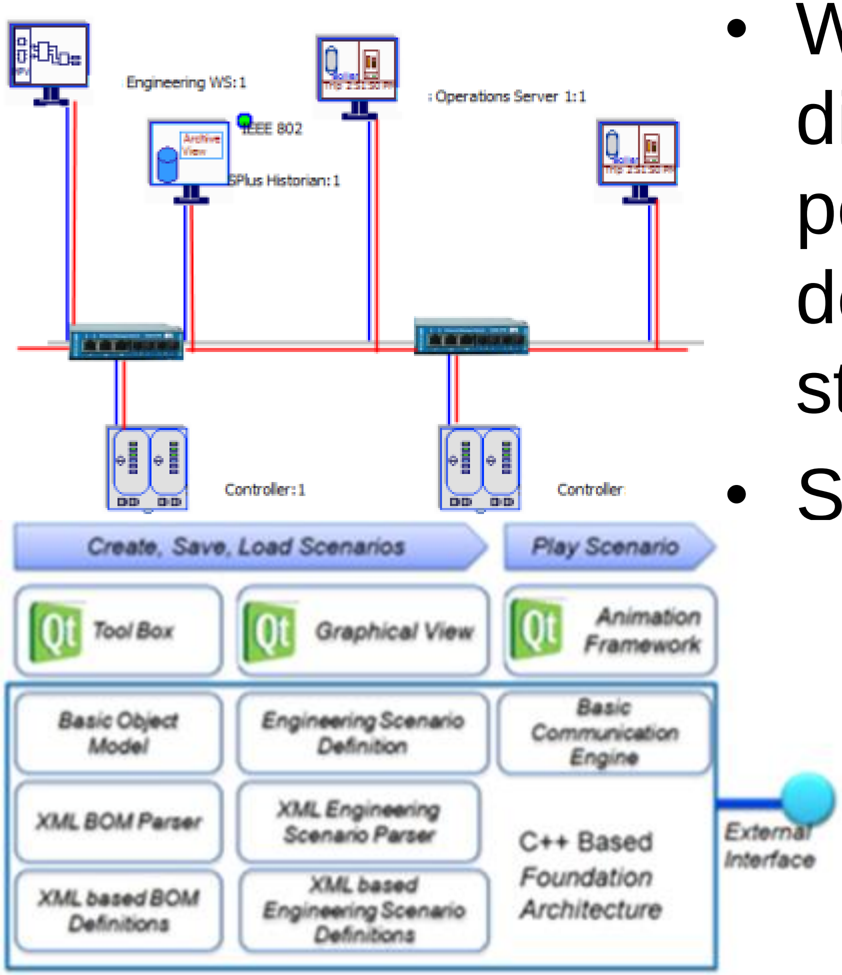
🟩 Solution Space

- Architecture Description is a key deliverable of the Architecture Definition
- Goals and Objectives of System are documented in consultation with stakeholders
- Architecture Description Demonstrates how the goals and objectives are achieved using a candidate architecture

Tools for Problem space

Runtime View

Development View



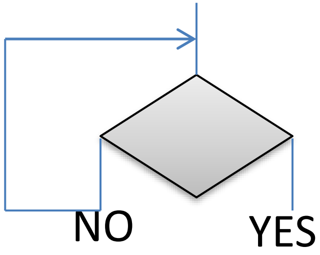
Engineering WS:1
IEEE 802
: Operations Server 1:1
SPlus Historian:1
Controller:1
Controller
Create, Save, Load Scenarios
Play Scenario
Tool Box
Graphical View
Animation Framework
Basic Object Model
Engineering Scenario Definition
Basic Communication Engine
XML BOM Parser
XML Engineering Scenario Parser
C++ Based Foundation Architecture
XML based BOM Definitions
XML based Engineering Scenario Definitions
External Interface

- We need a different view point and view defined based on stakeholders!
- Safety & Security
  'e perspectives

ViewPoints & Views help communicate better!

Tools for solution space

Is the solution suitable for the problem



- Architecture Patterns – POSA 1 and 2 [1]
- Architecture Tactics – Software Architecture in Practice [2]
- Design Patterns – GoF Design Patterns [3]
- Data structures and Algorithms in STL [4]

- Open Source /COTS/Reuse from within
- Reference Architectures

A Case Study of Architecture Definition

Systems Visualization Platform

User story 1:-Ability to define user scenario

User Profile	**User is product manager (Advanced knowledge of industrial automation components.)**
Background	The user is familiar of objects like HMI, Controller, Process Model. The user tries to document a requirement . The user would need this information be stored, shareable and executable by some other colleagues.
Objective	Able to position, instantiate the elements in a visual pane, connect them and define the sequence.
Narrative	The user should be able to define the connections between these elements using easy connectors. The user should be able to define the sequence of actions in the workflow. These aspects need to be stored file format
Acceptance Criteria	1. Elements HMI, Controller, Process Model are available and visible for instantiation 2. Elements can be dropped into editor pane and positioned 3. Elements can be connected together using communication elements 4. The topology and messages and their sequences can be described 5. The contents of the editor can be saved as a file and stored. 6. The contents are loaded and checked if they contain the full description and elements, connections and sequences.

Architecture Definition :- Applied

Architecturally Significant Use Cases

- Create, Save and Play Scenarios, communication with simulation system

Applying right patterns, tactics and arrive at Candidate Architecture

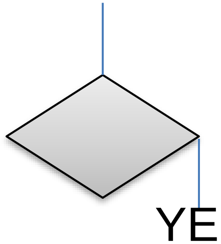
- Qt/.NET decision, Communication Stack decision

Guidance and Implementation

- Systems Explorer framework

Tools for solution space :- Applied

Is the solution suitable for the problem



YES

Architecture Patterns –Reflection, Layers

Architecture Tactics – Usability

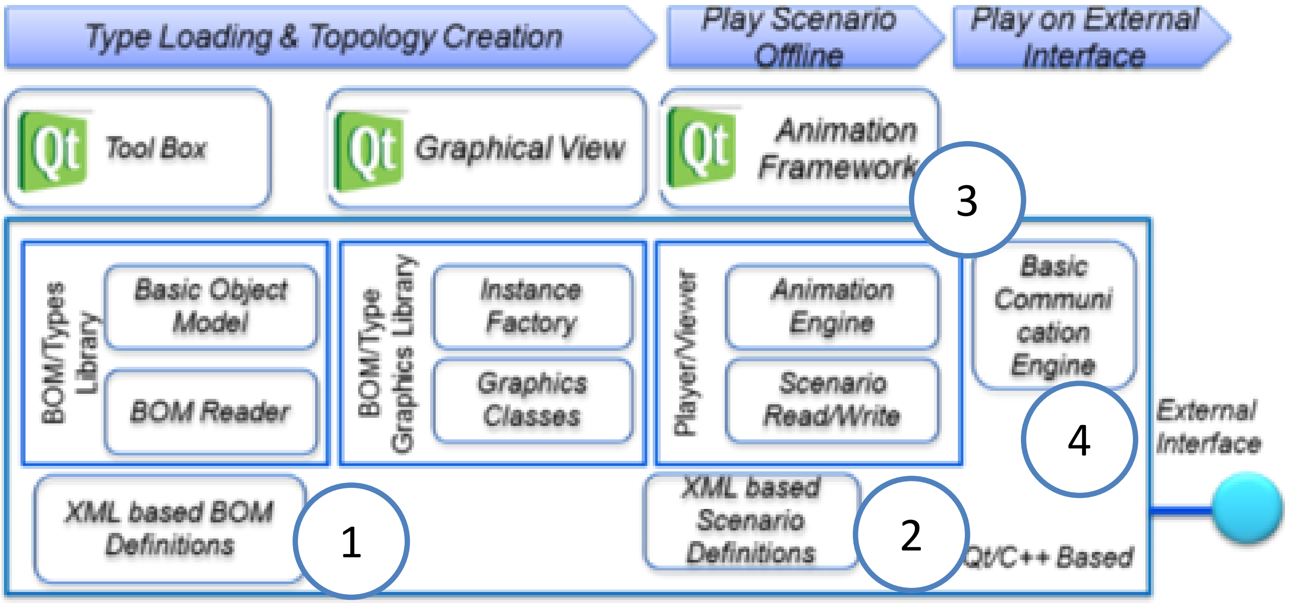
Design Patterns – Factory Method, Singleton

Data structures and Algorithms – XML Parsing

Open Source – Qt framework

Reference Architectures – ACE Framework

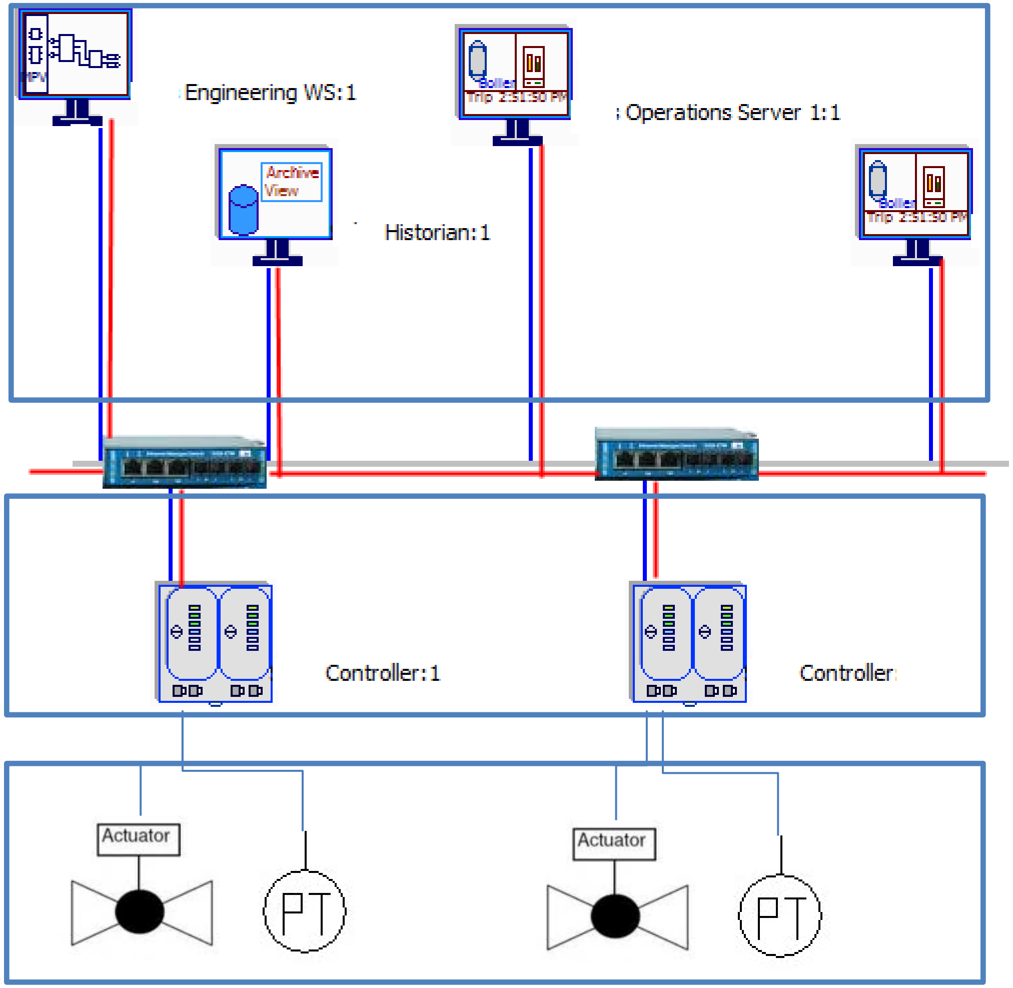
Architecture Description – Addressing View Point of Development Team



Layered View : *Describes the decomposition at a public level and interfaces*

1. Object Model & Toolbox
2. Engineering scenario definition and view
3. Animation Framework for scene play
4. Basic communication engine to interface with external simulation

Architecture Description – Addressing View Point of Product Manager



Automation View: Product Managers

1. Visualization Toolbox – What components available
2. How will a scenario look like?
3. How can we specify sequence of actions?
4. How can we save the scenario?

Documenting Key Technical Decisions

Date and Time	
Subject/Problem Description	
Assumptions	
Alternatives	
Decision/Solution Description	
Rationale	
Risks	
Participants	
Implications	

Technical Decisions Document

It is important to keep note of technical decisions

References

1. Pattern Oriented Software Architecture Volume 1 and 2 (POSA)
2. Len Bass , Paul Clements , Rick Kazman “Software Architecture In Practice” – Second Edition Apr 19, 2003
3. Erich Gamma , Richard Helm , Ralph Johnson, John Vlissides “Design Patterns: Elements of Reusable Object-Oriented Software” November 10, 1994 ,ISBN978-0201633610
4. Mark Nelson,"C++ Programmer's Guide to the Standard Template Library " December 1995
5. Nick Rozanski and Eóin Woods “Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives (2nd Edition)” Nov 4, 2011
6. Erich Gamma , Richard Helm , Ralph Johnson, John Vlissides “Design Patterns: Elements of Reusable Object-Oriented Software” November 10, 1994 ,ISBN978-0201633610
7. NIST SP 800-82, Guide to Industrial Control Systems (ICS) Security
8. European Commission & US National Science Foundation "Workshop Report and Recommendations - Engineering Software -Intensive Systems” –22-23 May 2004
9. Dr TV Gopal “Systems Engineering Essentials” , 27 March 2012
10. IEEE Magazine, “NextGen Vehicular Systems” , June 2012
11. Applications of Software Defined Networks in Industrial Automation – An Architectural analysis – Abhilash Gopalakrishnan
12. Requirements Engineering Visualization Platform- Architecture and Experiences – Abhilash Gopalakrishnan, Abhinna Biswal, Ashoka Shyamaprasad 2012.

Thank you!

- Discussion