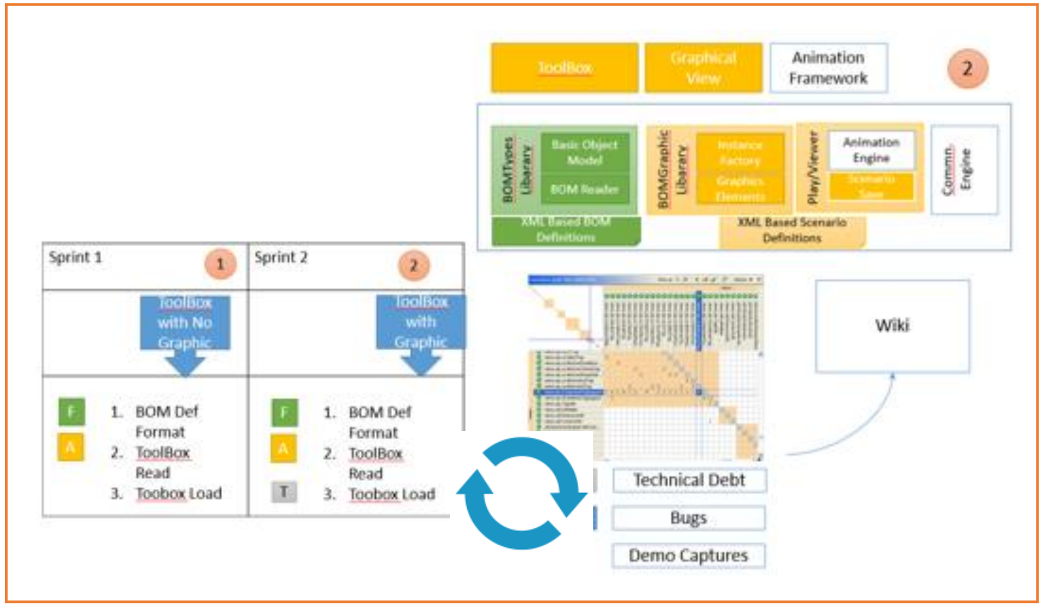
Software Architecture | In An Agile World

Presentation



Agenda

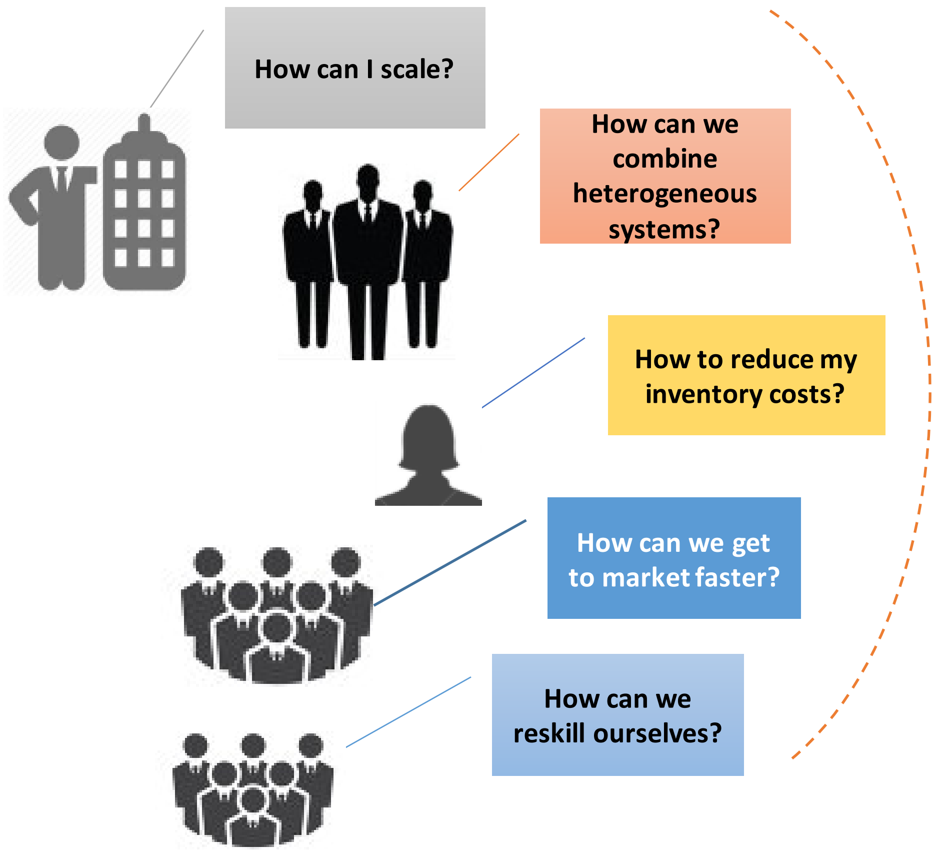
Need for Speed | Foundations | Roadmap | Agile Approach | Summary

Abhilash G, Principal Engineer, ABB/IDC/EPDA, 7 July 2018

Source:
Picture -

Software Architecture| Need for Speed

The Need



How can I scale?

How can we combine heterogeneous systems?

How to reduce my inventory costs?

How can we get to market faster?

How can we reskill ourselves?

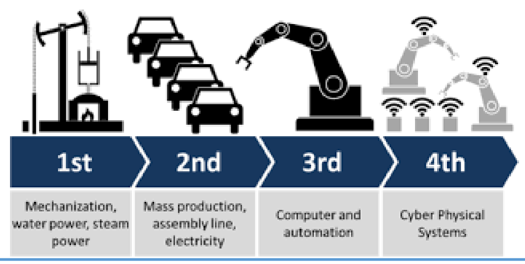
Forces from across the world driving revolutions!

Need for Scale

Need for Speed

Need for reduced infra.

Heterogeneous Systems


1st
2nd
3rd
4th
Mechanization, water power, steam power
Mass production, assembly line, electricity
Computer and automation
Cyber Physical Systems

Digital Transformation, Agile Transformation

The shift from centralized to more distributed styles is evident in the evolution happening in Smart Grids, Industry 4.0, Industrial internet and IoT, Society 5.0.

Drive speed by Industrialization

Drive Speed through R&D Cycle time reduction

Key Number

8.4 Billion Connected "Things" Will Be in Use in 2017

More devices than people- More M2M

Sources: Gartner, Japan.go.jp
http://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html
https://www.japan.go.jp/abenomics/_userdata/abenomics/pdf/society_5.0.pdf

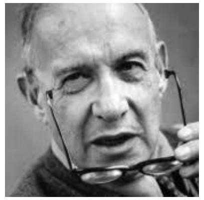
Technology is often like fashion-Distributed Architectures are back!

Software Architecture | Need for Speed

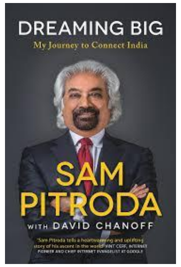
In new society

"What we need is an entrepreneurial society in which innovation and entrepreneurship are normal, steady and continuous." — Peter F. Drucker

https://hbr.org/2016/10/6-signs-youre-living-in-an-entrepreneurial-society

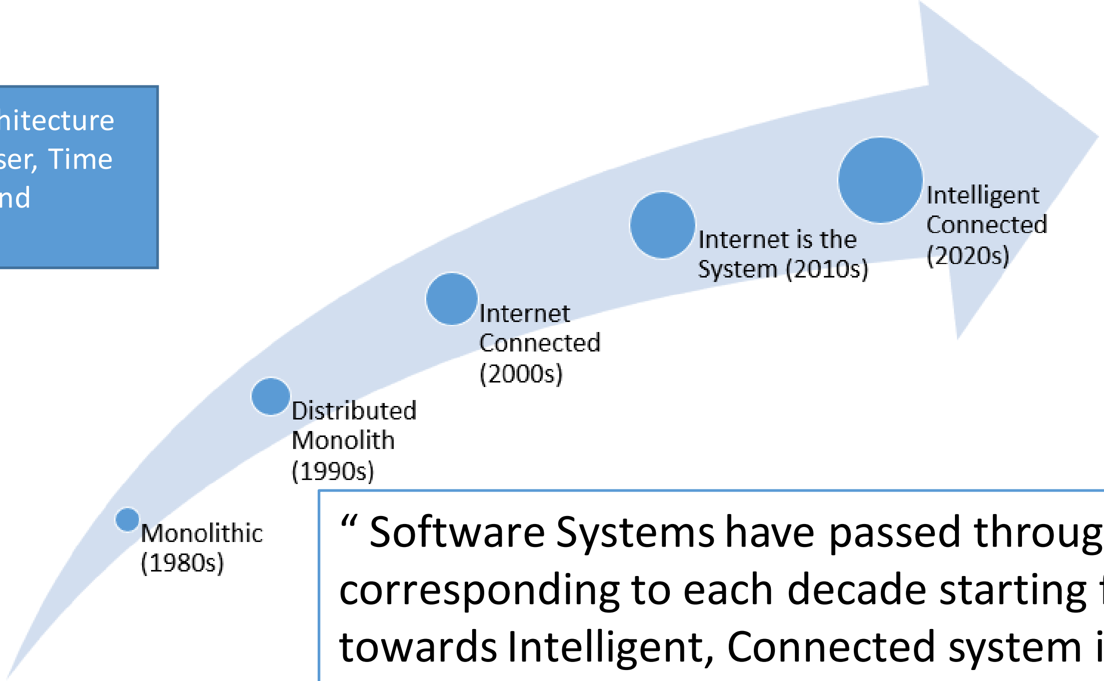


" And with a lot of arrogance and even more ignorance, I thought, Who better to fix it than me? This is something I need to do"

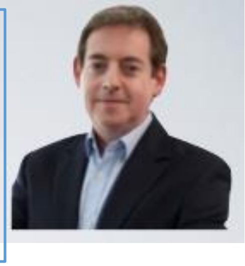


Budget predictability is 2-3x better with architecture practices. While Budget overrun 7 x less lesser, Time overrun 6 x less, Troubled projects 3x less and Customer satisfaction 1-2 points better

Survey among 49 projects Reported by Raymond Slot,PhDThesis,2010.


Monolithic (1980s)
Distributed Monolith (1990s)
Internet Connected (2000s)
Internet is the System (2010s)
Intelligent Connected (2020s)

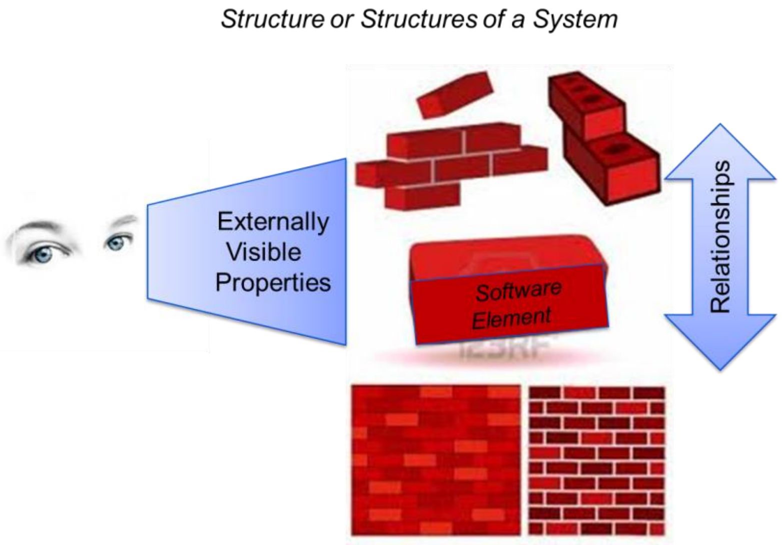
" Software Systems have passed through five ages corresponding to each decade starting from 1980 towards Intelligent, Connected system in 2020 " – Eoin Woods, CTO at Endava (prev Software Architect)



Who else can chart the roadmap like an Architect!

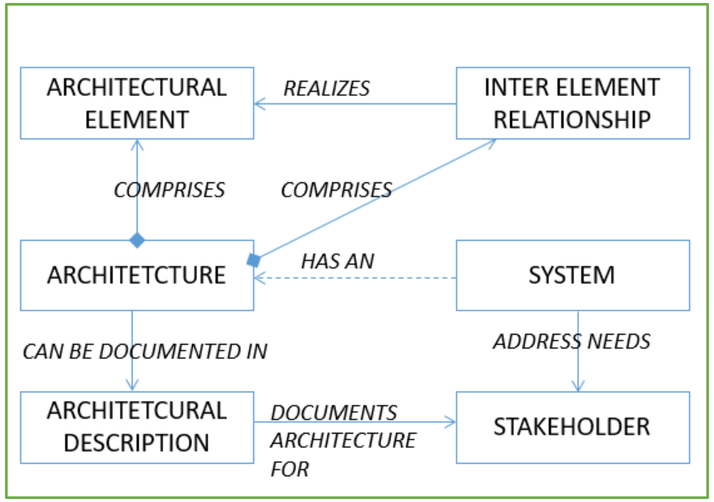
Software Architecture| Foundations

Definitions and Standards

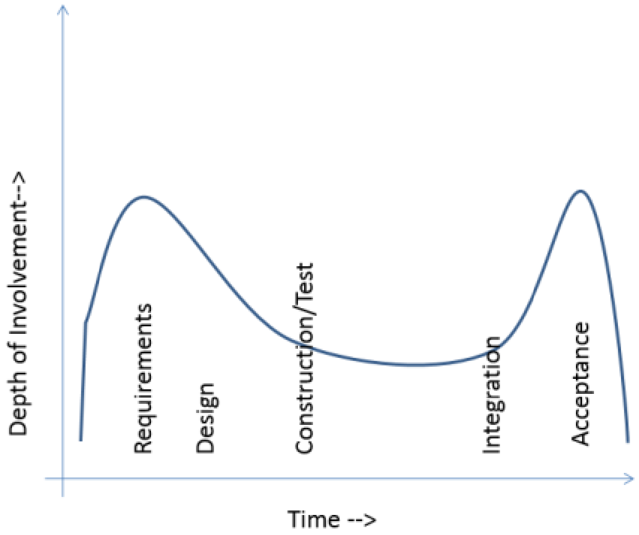

Structure or Structures of a System
Externally Visible Properties
Software Element
Relationships

Software Architecture of a system is the structure of structures of a system with its externally visible properties and relationships

IEC/ISO/IEEE 42010



A system is built to address the needs, concerns, goals and objectives of its stakeholders. The architecture of system comprises of its architectural elements and their interrelationships. Architecture is documented in Architecture Description and demonstrates that it has met their needs.

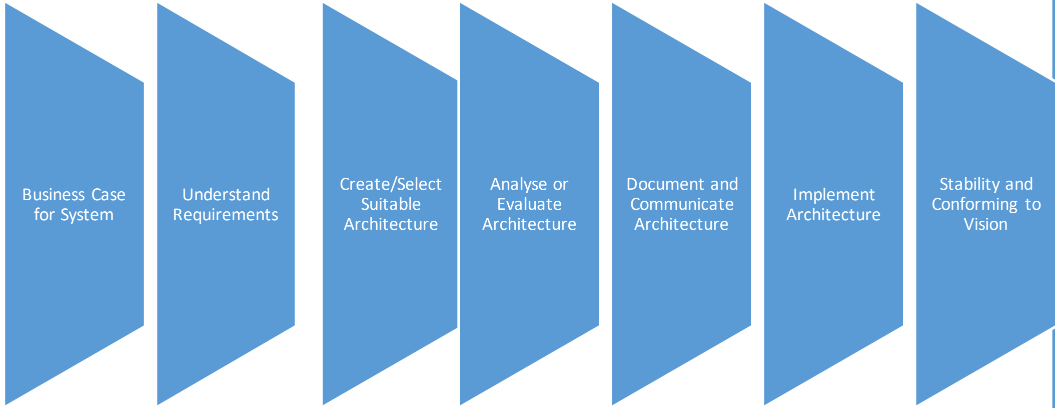


Depth of involvement of architect could vary across phases in the life cycle. Architect uses programming skills to build demonstrators showing how requirements are met.

'Architecture is about the important stuff. Whatever that is' – Martin Fowler

Software Architecture| Foundations

Skills, Steps Deliverables


Business Case for System
Understand Requirements
Create/Select Suitable Architecture
Analyse or Evaluate Architecture
Document and Communicate Architecture
Implement Architecture
Stability and Conforming to Vision

Deliverables

1. A clear complete, consistent and attainable set of goals (stress functional goals)
2. A description of the broader context of the system including global standards.
3. A concept of the system
4. A concept of operations of the system including contingency & emergency operations
5. A functional decomposition of the system with at least 2 layers of decomposition with primary and secondary functions, process flow , supporting and interface processes
6. The decomposition of form to two levels of detail and allocation of functions to form with details of all external interfaces and interface control process and a notion of development cost, schedule and risk

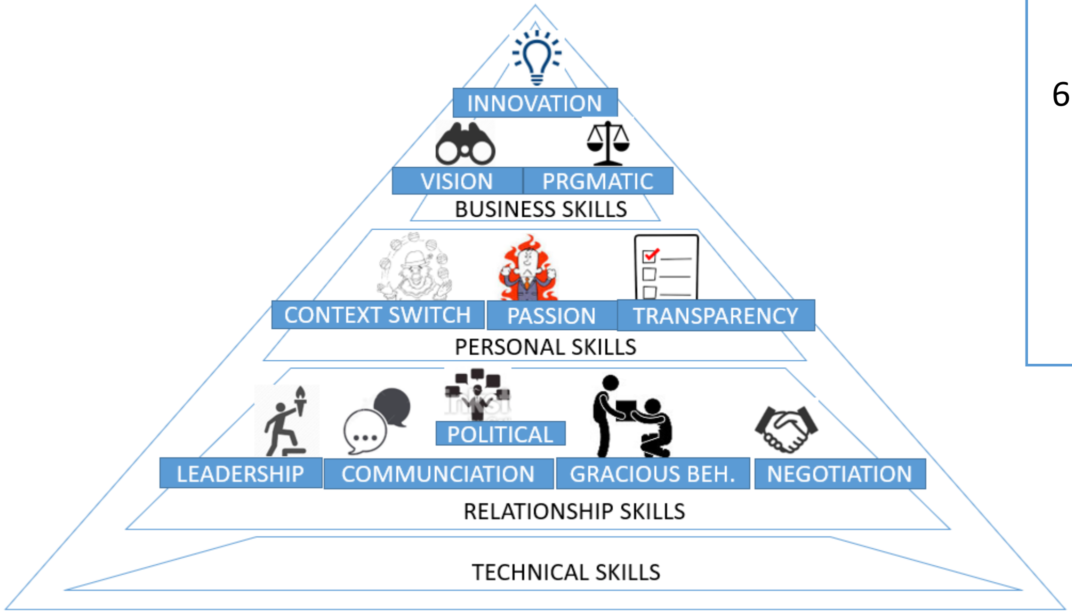
Architecture Description

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

- Solution: Achieving the Goals and System Objectives

🟦 Problem Space

🟩 Solution Space



Why	What	How	When	Who	How much
•**The System is built - Need**	•The System accomplishes - Goals	•The System acts - Function	•The elements are - Form	•Does them – Operator	•Does it cost?

The deliverables span across the phases and many are early decisions!

Software Architecture| Foundations

A Great Reference Architecture

WWW

Berners-Lee "Web's major goal was to be a shared information space through which people and machines could communicate."

What was needed was a way for people to store and structure their own information, whether permanent or ephemeral in nature, such that it could be usable by themselves and others, and to be able to reference and structure the information stored by others so that it would not be necessary for everyone to keep and maintain local copies. The intended end-users of this system were located around the world, at various university and government high-energy physics research labs connected via the Internet. Their machines were a heterogeneous collection of terminals, workstations, servers and supercomputers, requiring a hodge podge of operating system software and file formats. The information ranged from personal research notes to organizational phone listings. The challenge was to build a system that would provide a universally consistent interface to this structured information, available on as many platforms as possible, and incrementally deployable as new people and organizations joined the project.

Hall of Fame

The Internet – HTTP Specification – Apache HTTP Server Project

Performance	Scalability	Simplicity	Modifiability	Visibility	Portability	Reliability
• **Network Performance** • **User Perceived Performance** • **Network Efficiency**	• Ability to support large number of components and transactions	• Complexity, Understandability • Verifiability	• Evolvability • Extensibility • Customization • Configuration • Reusability	• Ability of component to monitor or mediate between two components	• Can run in different environments	• Degree of susceptibility to failure in case of partial failure from components

Low Entry Barrier	Relationships	Presentation	Control	Anarchiac Scalability	URI, HTTP, HTML
HYPERMEDIA	Links	DISTRIBUTED HYPERMEDIA		Independent Deployment	

Software Architecture| RoadMap

The Approach

" Who wants an Architect" – Martin Fowler

Architecting On the Go

*'There's speed work, and then there's speed work. When most runners talk about doing speed work, they mean things like mile repeats at 10K race pace, or a set of fast 200s, or maybe even a 5-mile tempo run. Such workouts, of course, are integral to becoming a faster runner. But they're not really speed work, **if by "speed" we mean the fastest** you can run for a very short distance. When **I talk about speed, I mean your maximal velocity**—your **top speed—which even world-class sprinters can sustain for no more than 30-40m**.'*

***Jay Johnson** in https://www.runnersworld.com/advanced/a20788111/speed-development/*

"Architecting as Risk and Cost Discipline." — Eltjo Poort

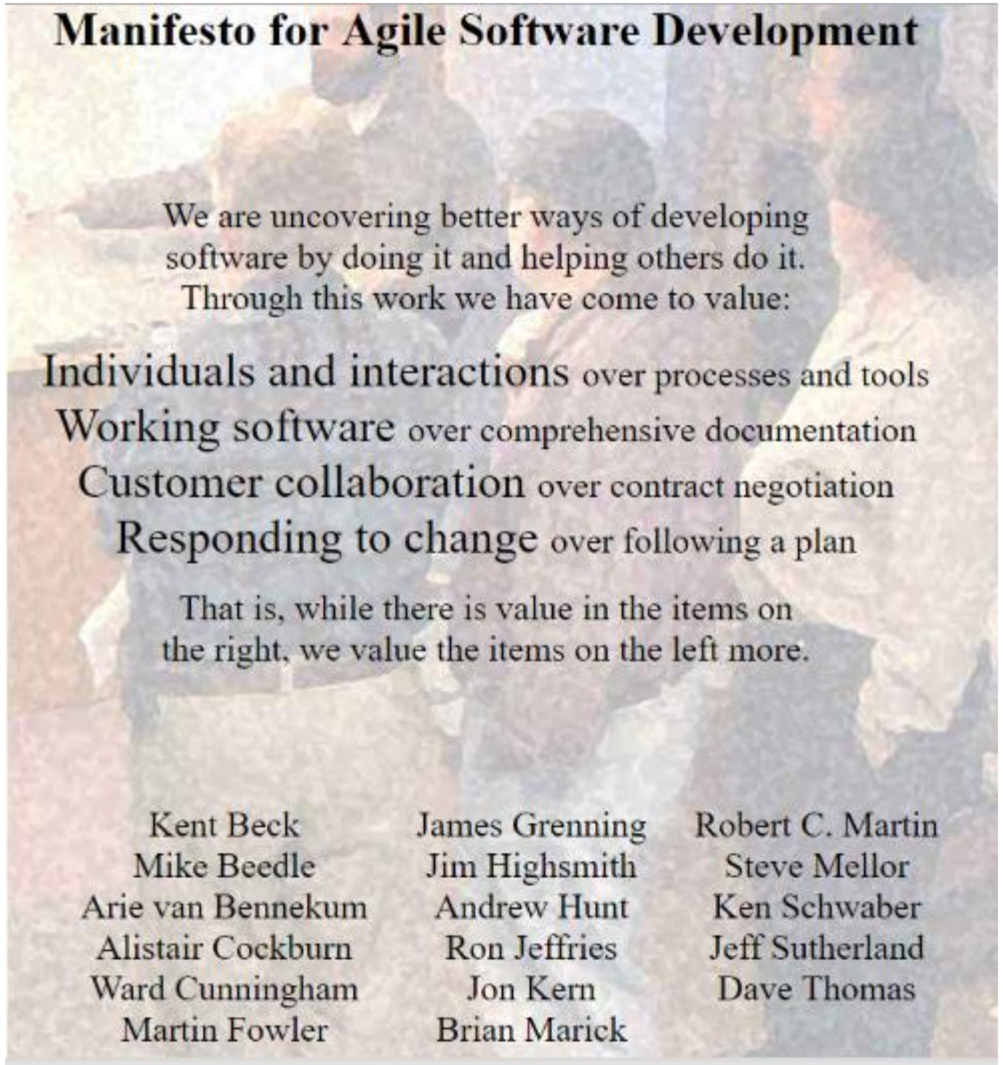
"R&D As Experimentation System" — Jan Bosch

" A fairly recent evolution is
-Architect acts less upfront design of structures i.e., Significant decisions made just in time
-Architect deals with more probability than certainty

- Large Systems Policy Driven Automation
- Architecture is still very much art of possible – financial constraints like cloud pricing in consideration
- Radical intelligence , Dynamic Components, Cloud Platform Deployment, Connection to things in mainstream" -Eoin Woods

Software Architecture| Agile

Foundations



Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

Architecting On the Go

“The best architectures, requirements, and designs emerge from self-organizing teams.”

Risk and Cost Based Architecting Principles

1. *Decisions are your main deliverable.*
2. *Keep a backlog of architectural concerns.*
3. *Let economic impact determine your focus.*
4. *Keep it small.*
5. *Use Just Enough Anticipation.*

Applying RCDA in our context we have arrived at :

1. *Prepare an Architecture Vision*
2. *Prepare a Decomposition*
3. *Identify the Most Significant Elements*
4. *Arrive at Risk and Cost Based Roadmap*
5. *Feedback, Analyzing Progress*
6. *Communicating the Progress*

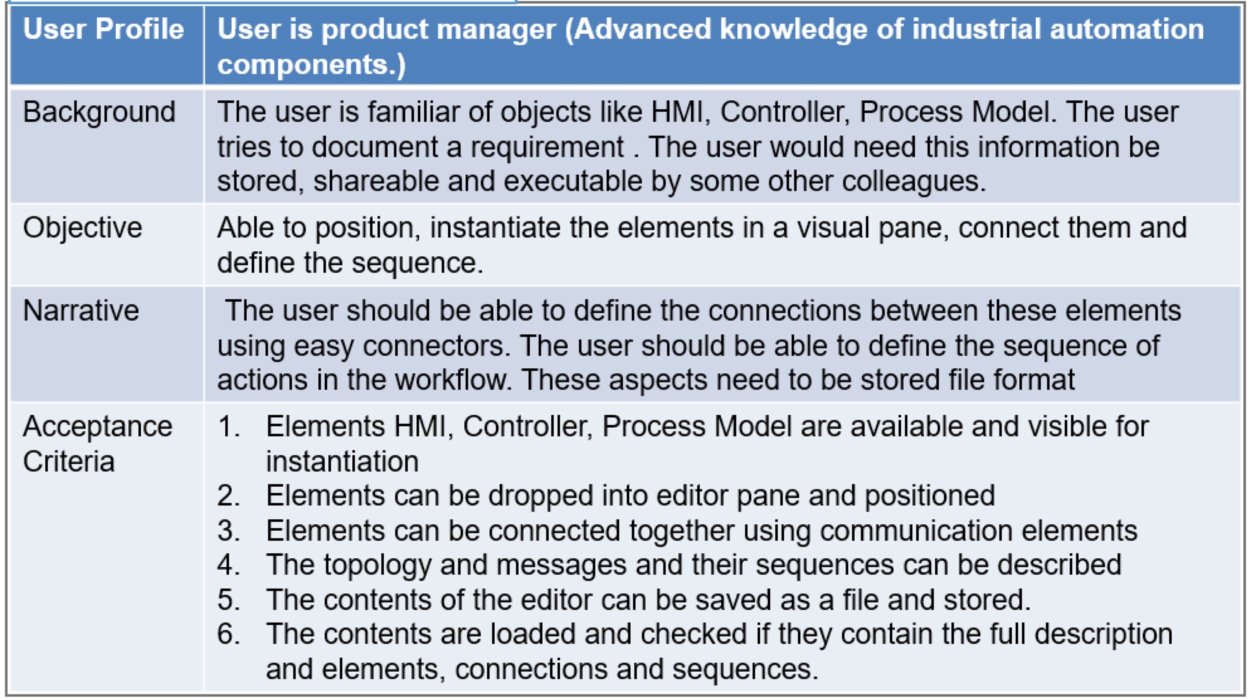
Software Architecture| Agile Approach

Insights into Scope

Architecting On the Go

One day Alice came to a fork in the road and saw a Cheshire cat in a tree.
'Which road do I take?' she asked.
'Where do you want to go?' was his response.
'I don't know', Alice answered.
'Then', said the cat, 'it doesn't matter'.

Overview of Scope



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.

Reference Standard

Simulation Interoperability Standards Organization (SISO)

SISO-STD-003-2006: Standard for Base Object Model (BOM) Template Specification ***(8 May 06)***

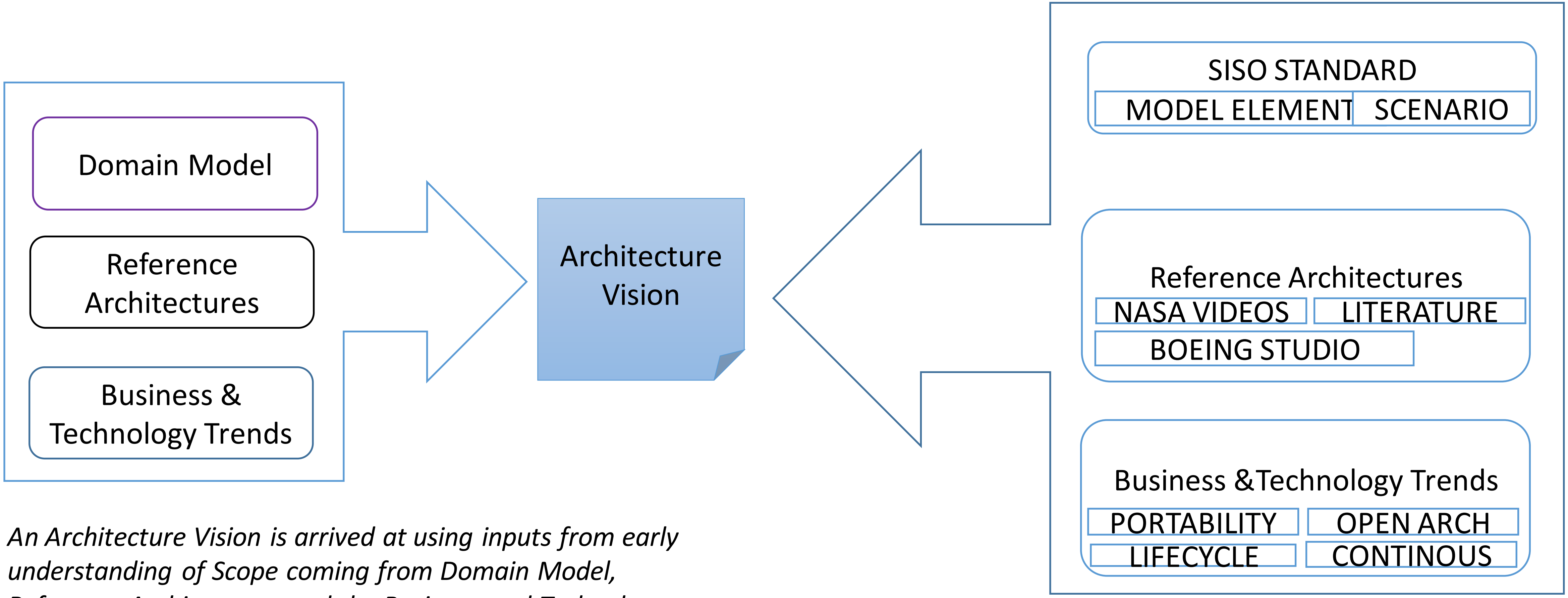
SISO-STD-007-2008: Standard for Military Scenario Definition Language (MSDL) (*reaffirmed 11 May 2015*)

Need some insights into scope

Software Architecture| Agile Approach

Prepare an Architecture Vision

Architecting On the Go



Domain Model

Reference Architectures

Business & Technology Trends

Architecture Vision

SISO STANDARD

MODEL ELEMENT | SCENARIO

Reference Architectures

NASA VIDEOS | LITERATURE

BOEING STUDIO

Business &Technology Trends

PORTABILITY | OPEN ARCH

LIFECYCLE | CONTINOUS

An Architecture Vision is arrived at using inputs from early understanding of Scope coming from Domain Model, Reference Architectures and the Business and Technology Trends.

Architectural Vision includes Models, Ref Architectures & Trends

A Target State is required to start !

Software Architecture | Agile Approach

Early Decisions

Architecting On the Go

Element	Selection	Rationale
Source Control	GITHUB	Lifecyle
Language and Framework	C++, QT, XML	Portability
Standardized	SISO	Open Architecture
Build System	Cmake, Jenkins, CPPUTest	Open , Easy to Use

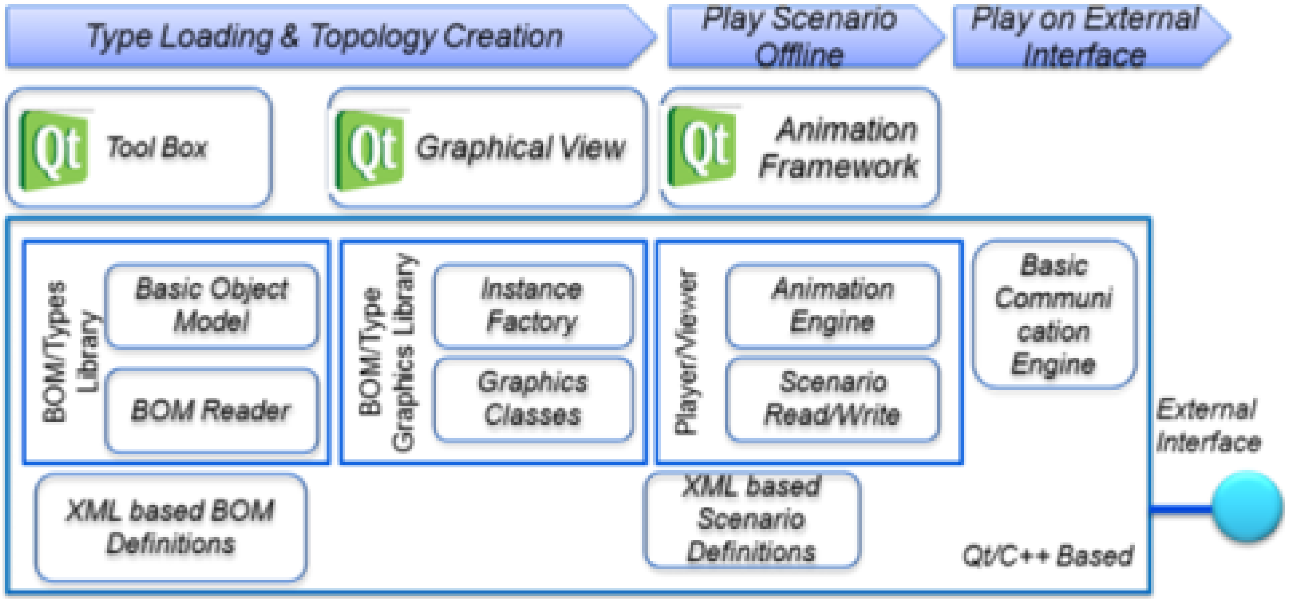
Some early decisions on Infrastructure need to be taken in the very beginning

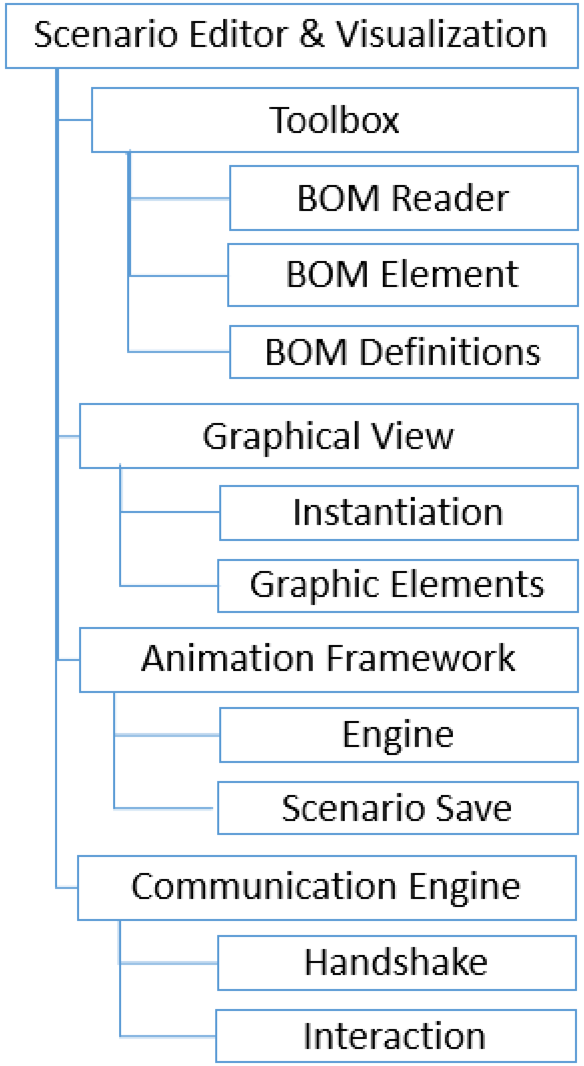
Early Decisions have major impact!

Software Architecture | Agile Approach

Decomposition and Layered View

Architecting On the Go




- Scenario Editor & Visualization
  - Toolbox
    - BOM Reader
    - BOM Element
    - BOM Definitions
  - Graphical View
    - Instantiation
    - Graphic Elements
  - Animation Framework
    - Engine
    - Scenario Save
  - Communication Engine
    - Handshake
    - Interaction

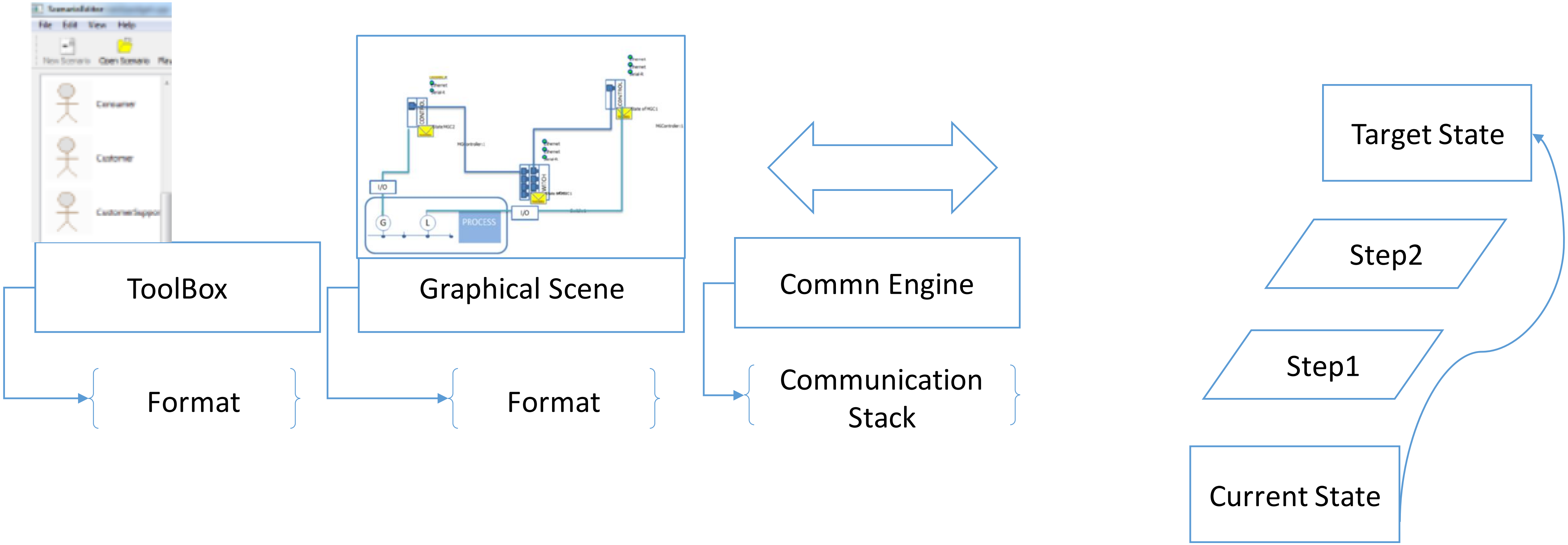
Based on the Domain Model and Reference Architecture and Prepare a Decomposition as much as we know

Decomposition based on Reference Architectures

Software Architecture| Agile Approach

Most Significant Elements

Architecting On the Go



The Most significant elements identified considering the Key Interfaces, Key Interactions and based on experience and proven knowhow which elements are risky and could be costly to make a change later. This should guide us on performing architecture evaluation or checking fulfillment of functional and non functional requirements later.

Identify the Most significant elements

Software Architecture| Agile Approach

Risk Based Roadmap

Architecting On the Go

Sprint 1	Sprint 2	Sprint 3	Sprint 4
ToolBox with No Graphic	ToolBox with Graphic	Scenario Creation & Save	STABILIZE
F A 1. BOM Def Format 2. ToolBox Read 3. Toolbox Load	F A T 1. BOM Def Format 2. ToolBox Read 3. Toolbox Load	F A T 1. Scenario Format 2. Information Flow 3. Connection & Save	F A T 1. BOM Def Format 2. ToolBox Read 3. Toolbox Load 4. Scenario Save

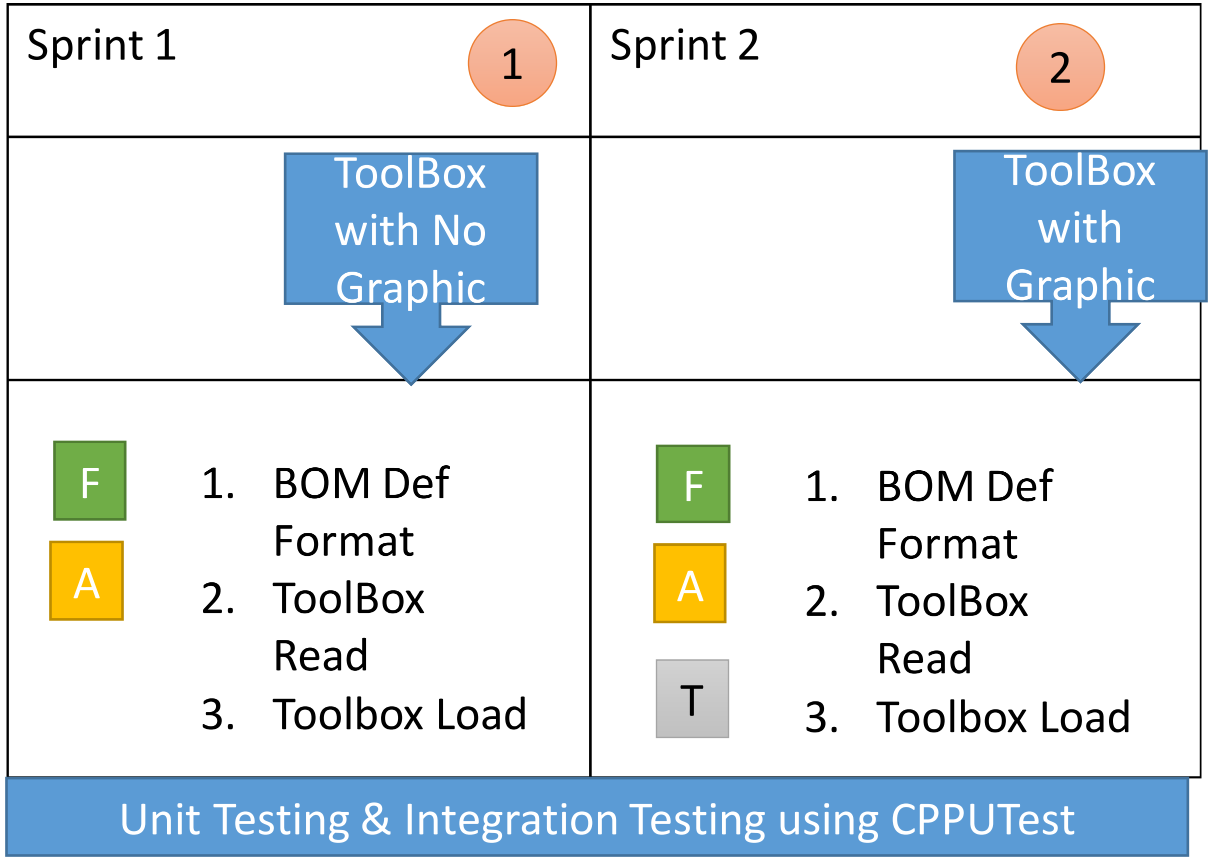
The dependencies are controlled. Architectural Elements, Features and Technical Debt Handling all go into the Backlog

Roadmap Based on Target Architecture Vision, Decomposition and elements

Software Architecture| Agile Approach

Measuring Progress

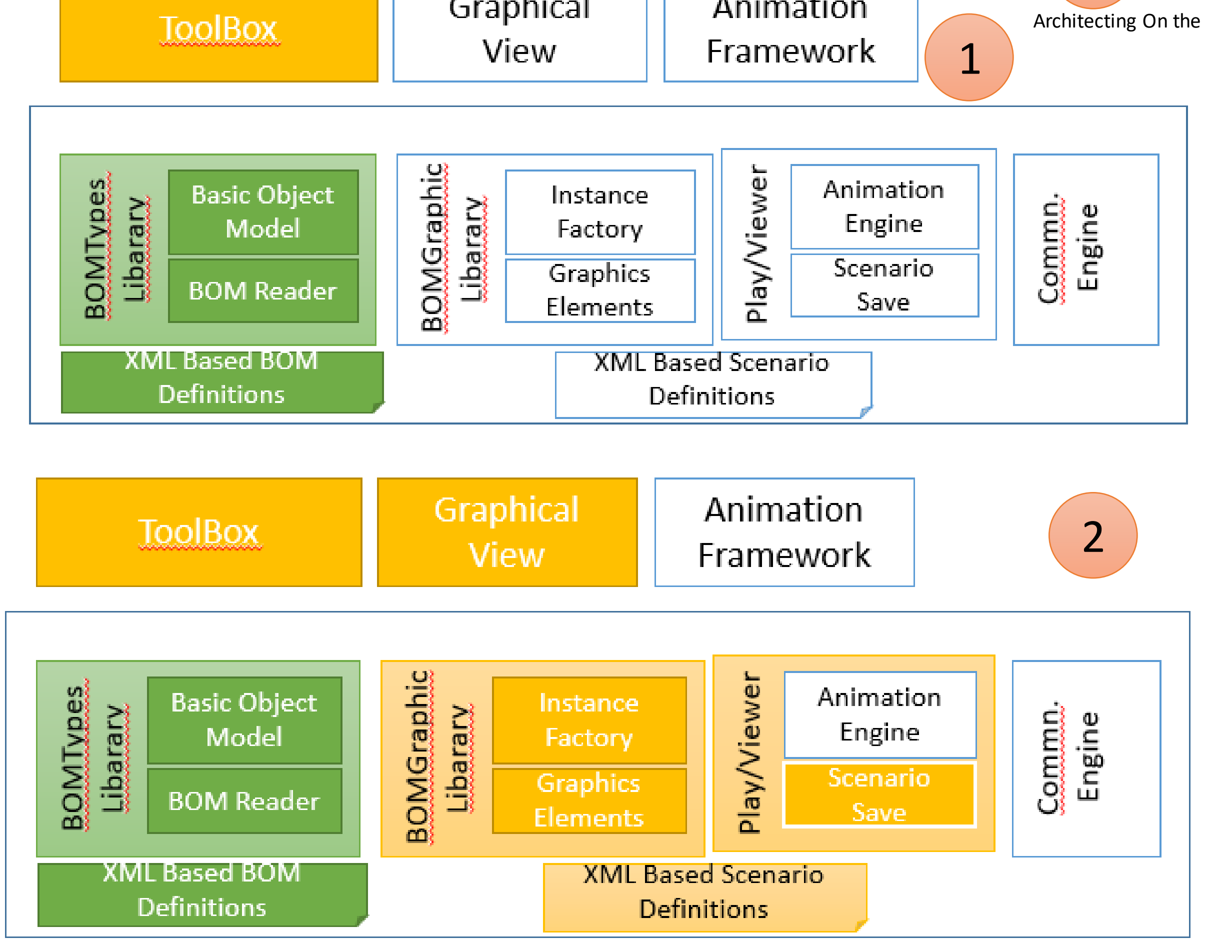
Architecting On the Go



1. BOM Def Format
2. ToolBox Read
3. Toolbox Load

1. BOM Def Format
2. ToolBox Read
3. Toolbox Load

Unit Testing & Integration Testing using CPPUTest



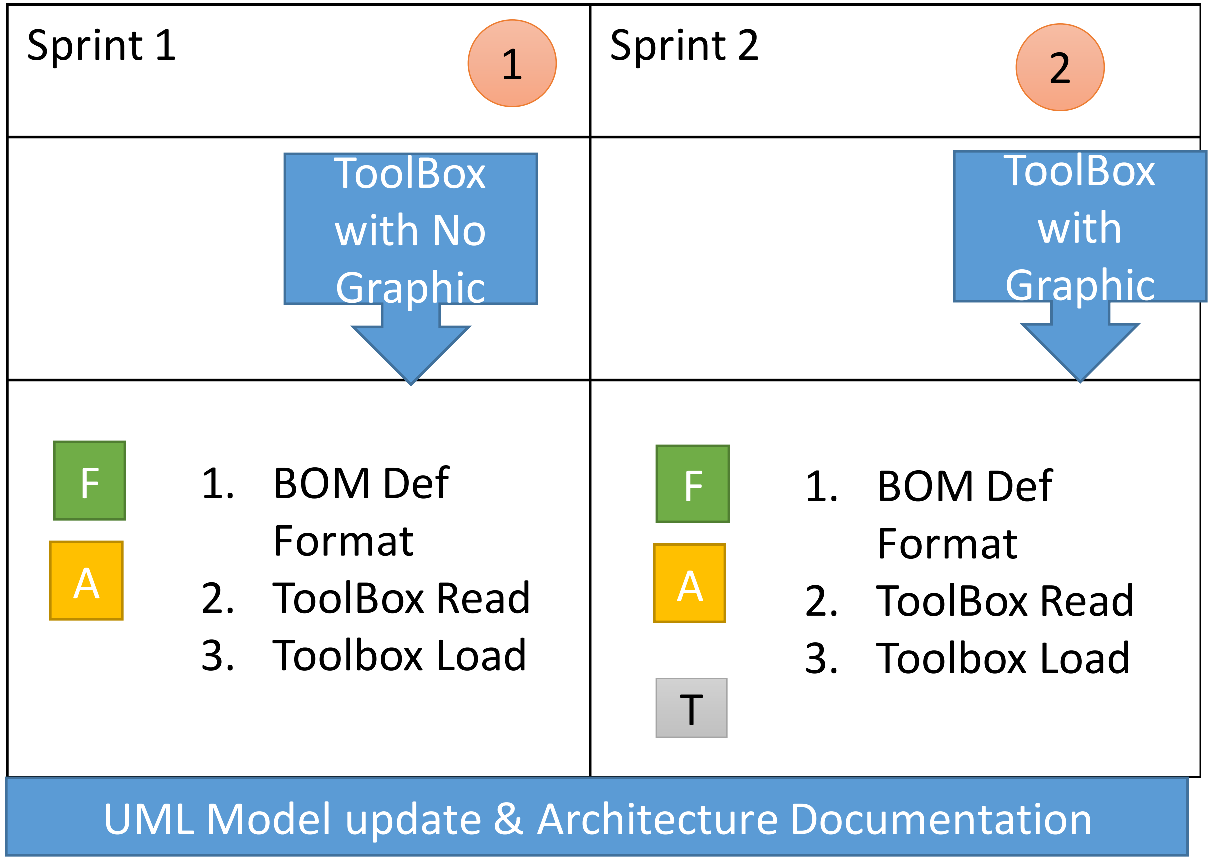
Measuring the Progress is done by pulling the Iteration result, performing Structure 101 analysis Performing Integration Testing and Demo testing and then marking it as ready.

Feedback Loop to Measure Progress

Software Architecture| Agile Approach

Communicating the Progress

Architecting On the Go



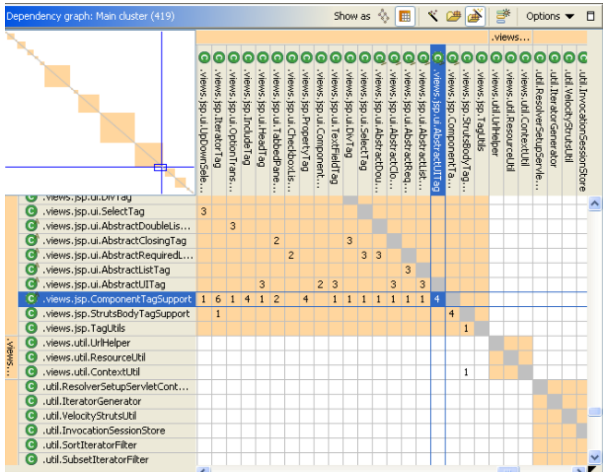
Sprint 1:

1. BOM Def Format
2. ToolBox Read
3. Toolbox Load

Sprint 2:

1. BOM Def Format
2. ToolBox Read
3. Toolbox Load

UML Model update & Architecture Documentation



																					.views…						
		.views.jsp.ui.UpDownSele…	.views.jsp.IteratorTag	.views.jsp.ui.OptionTrans…	.views.jsp.IncludeTag	.views.jsp.ui.HeadTag	.views.jsp.ui.TabbedPane…	.views.jsp.ui.CheckboxLis…	.views.jsp.PropertyTag	.views.jsp.ui.Component…	.views.jsp.ui.TextFieldTag	.views.jsp.ui.DivTag	.views.jsp.ui.SelectTag	.views.jsp.ui.AbstractDou…	.views.jsp.ui.AbstractClo…	.views.jsp.ui.AbstractReq…	.views.jsp.ui.AbstractList…	.views.jsp.ui.AbstractUITag	.views.jsp.ComponentTa…	.views.jsp.StrutsBodyTag…	.views.jsp.TagUtils	.views.util.UrlHelper	.views.util.ResourceUtil	.views.util.ContextUtil	.util.ResolverSetupServle…	.util.IteratorGenerator	.util.VelocityStrutsUtil	.util.InvocationSessionStore
	.views.jsp.ui.DivTag																											
	.views.jsp.ui.SelectTag	3																										
	.views.jsp.ui.AbstractDoubleLis…			3																								
	.views.jsp.ui.AbstractClosingTag						2					3																
	.views.jsp.ui.AbstractRequiredL…							2					3	3														
	.views.jsp.ui.AbstractListTag															3												
	.views.jsp.ui.AbstractUITag					3				2	3				3		3											
	.views.jsp.ComponentTagSupport	1	6	1	4	1	2		4		1	1	1	1	1	1	1	4										
	.views.jsp.StrutsBodyTagSupport		1																4									
	.views.jsp.TagUtils																			1								
.views…	.views.util.UrlHelper																											
	.views.util.ResourceUtil																											
	.views.util.ContextUtil																			1								
	.util.ResolverSetupServletCont…																											
	.util.IteratorGenerator																											
	.util.VelocityStrutsUtil																											
	.util.InvocationSessionStore																											
	.util.SortIteratorFilter																											
	.util.SubsetIteratorFilter																											

Wiki

T	Technical Debt
B	Bugs
	Demo Captures

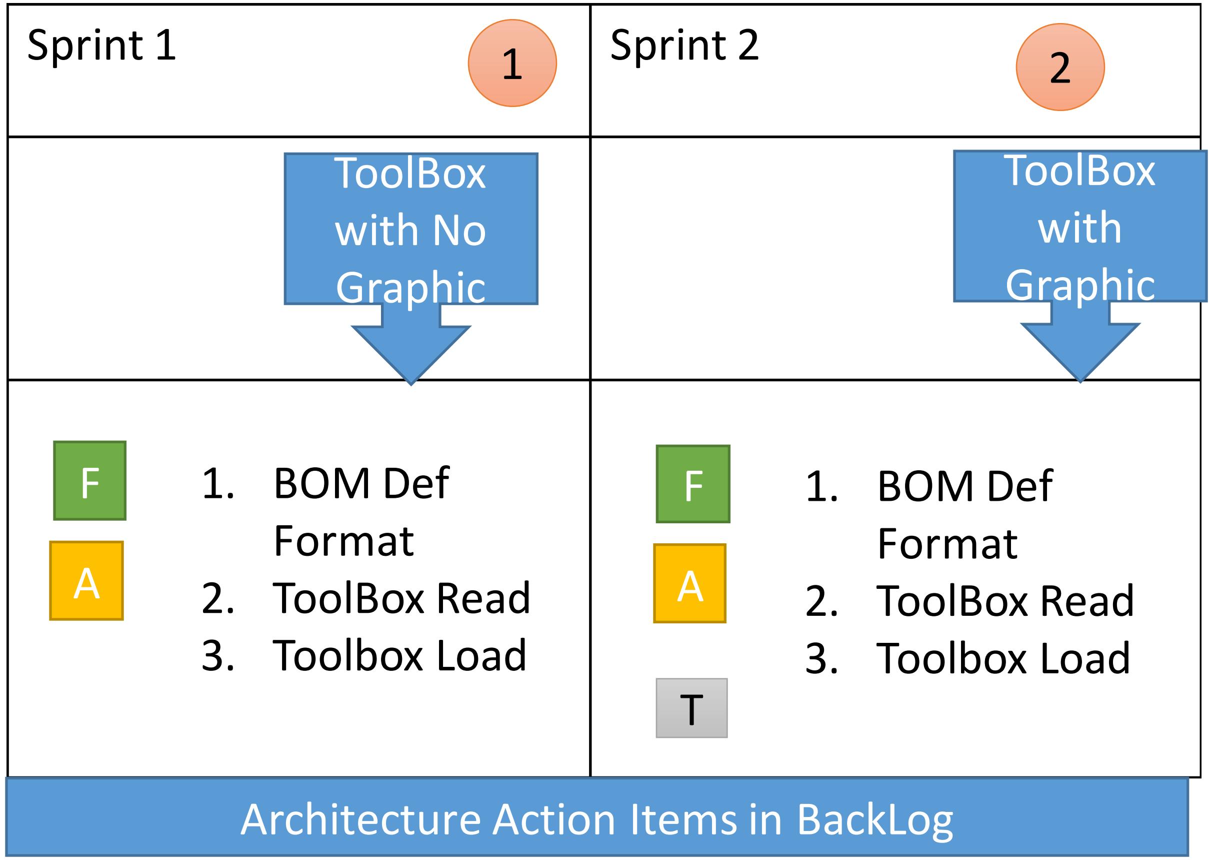
Communicating Involves again Sprint on Sprint Measurement and Results including the interfaces updated, Architecture Updates all updating in Wiki as part of Continuous Communication Strategy.

Continuous Communication of Progress!

Software Architecture| Agile Approach

Architectural elements Grooming

Architecting On the Go



Sprint 1

1

ToolBox with No Graphic

F

A

1. BOM Def Format
2. ToolBox Read
3. Toolbox Load

Sprint 2

2

ToolBox with Graphic

F

A

T

1. BOM Def Format
2. ToolBox Read
3. Toolbox Load

Architecture Action Items in BackLog

Architecture Description

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

- Solution: Achieving the Goals and System Objectives

🟦 Problem Space

🟩 Solution Space

Structure 101

CAFFEA Framework involving Chief Architect, Governance Architects Refactoring

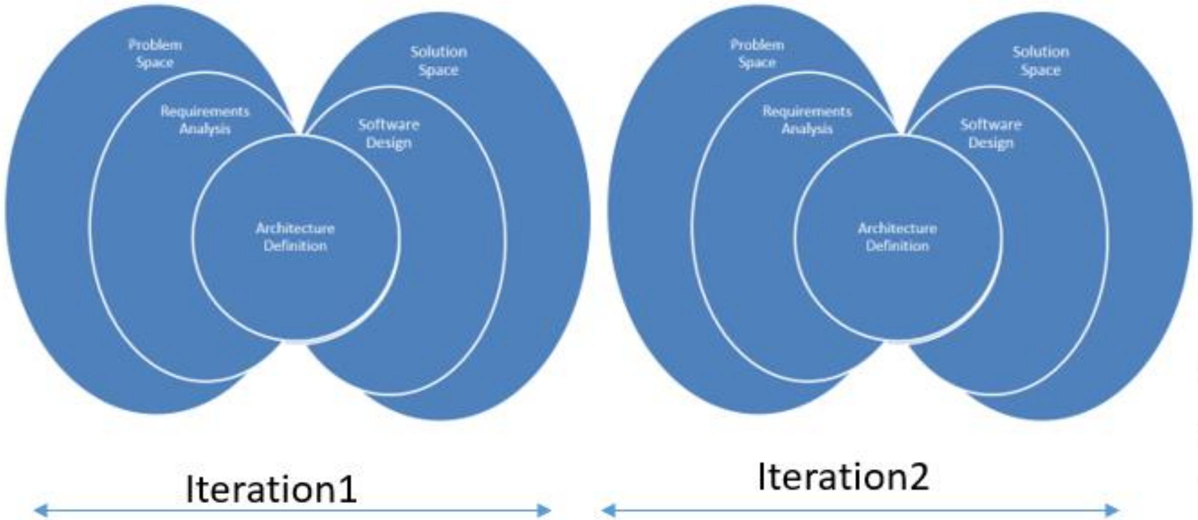
Architectural elements get continuously groomed as well as part of Grooming of Backlog items. The idea Is to check the Goals and System Objectives and based on the analysis of progress as well as Architectural Document updates, Keep analyzing the Gaps.

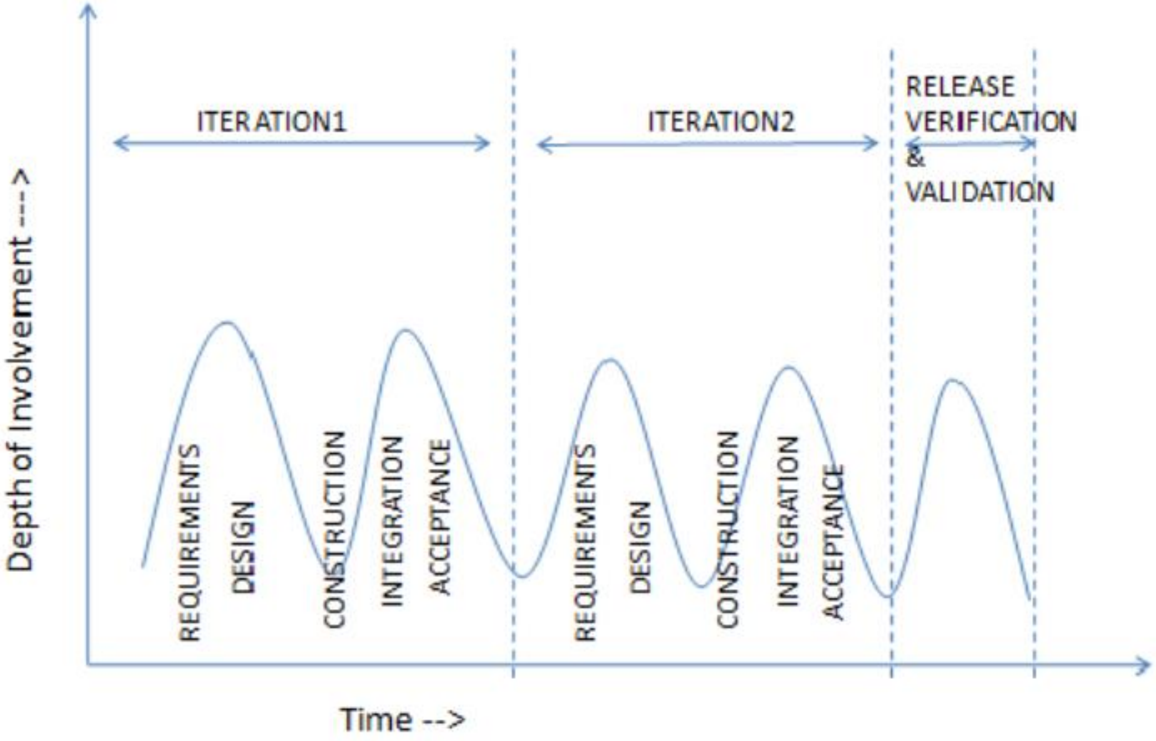
Architectural elements analysis as part of grooming

Software Architecture| Summary

“Good programmers know what to write. Great ones know what to rewrite (and reuse)..” – ***Eric S. Raymond****- The Cathedral & The Bazaar*

Architecting On the Go





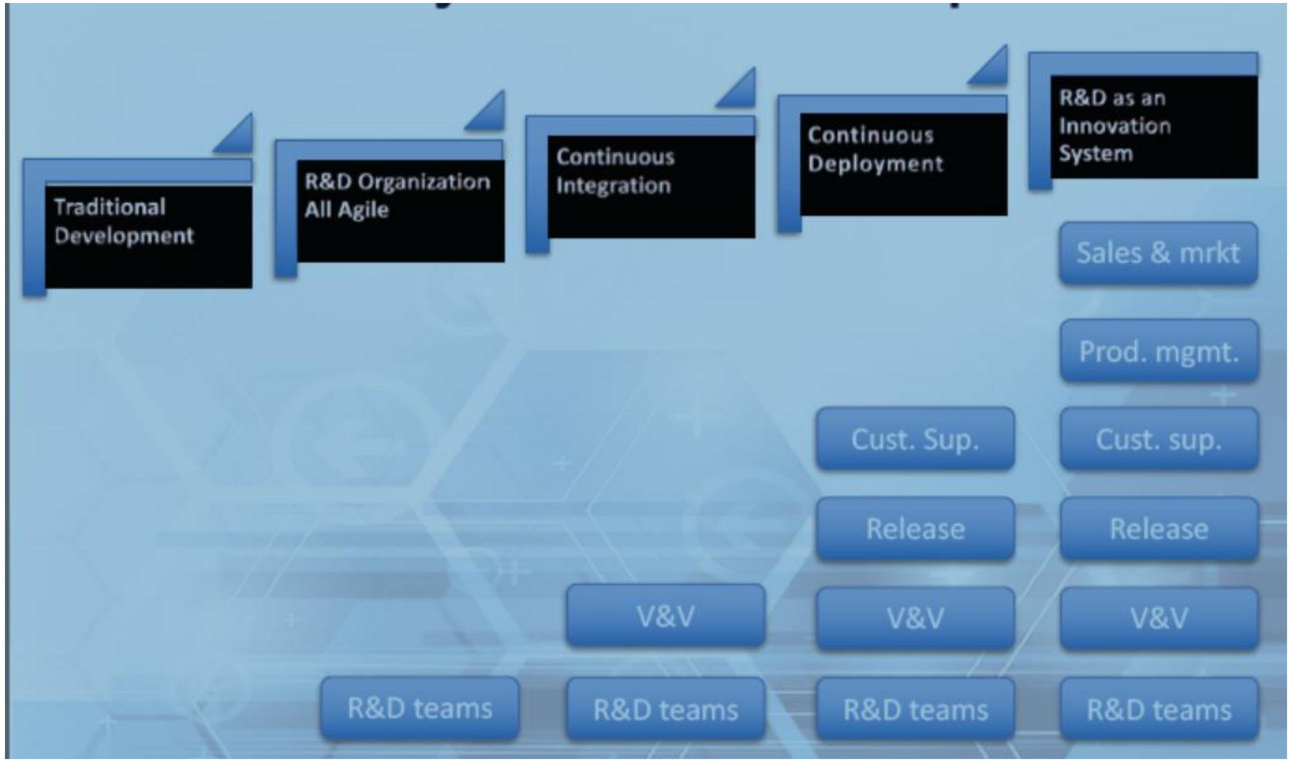
Originally Architect is Customer’s Person. Journey Maps and Innovation to delight customer is important!

Architect is a development team’s person when involved in design. Teams need Architects support for boosting morale, end to end execution, since they model themselves on servant leadership..

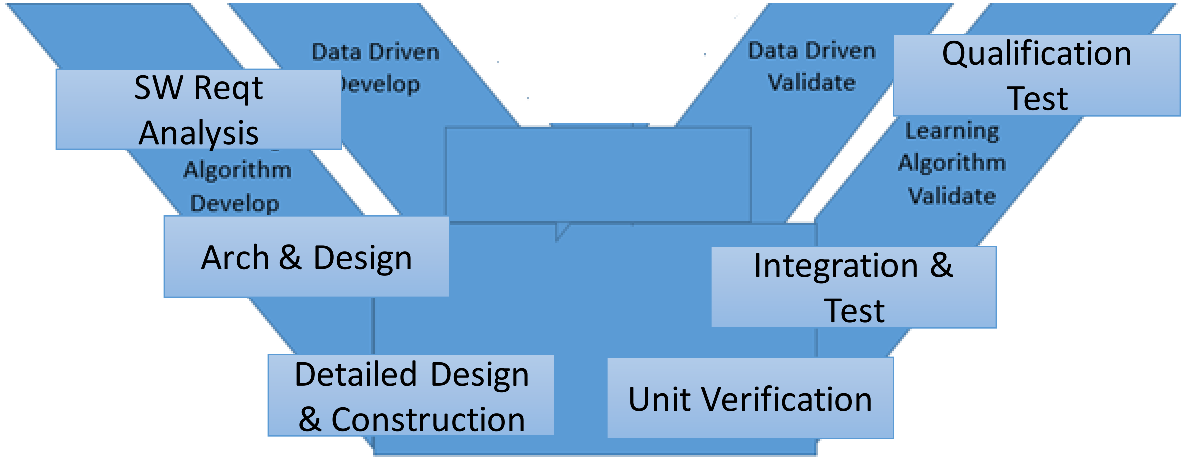
Depth of involvement of architect is higher and continuous. Need to drive a Continuous Integration and Continuous Test Strategy in addition to support to team wherever required.

Software Architecture| Summary

Architecting On the Go



Stairway to Heaven – R&D as Innovation System



Automotive SPICE 3.0
http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdf

Future Software Engineering Seems to look forward to combining machine learning algorithm development, data driven development, model driven development and their validation involved over a continuous chain.

SW Architecture| References

Books and Material

Architecting On the Go

1. Eon Woods, “Software Architecture in A Changing World”, IEEE Software Nov 2016.
2. Eltjo Poort, “Adapting Architecture Practices to Changing Times to Why and Back Again”, SATURN, May 2016.
3. Martin Fowler, “Who Needs an Architect?”, IEEE Software , 2003.
4. Dave Hendricksen, “12 Essential Skills for Software Architects”.
5. ISO/IEC/IEEE 42010:2011(E) - ISO/IEC/IEEE Systems and software engineering -- Architecture description.
6. Nick Rozanski and Eoin Woods, “Software Systems Architecture: Working with Stakeholders Using Viewpoints .”
7. Bruce Cameron, Edward Crawley, Daniel Selva, “System Architecture, Global Edition”.
8. Roy Fielding, ‘Architectural Styles and the Design of Network-based Software Architectures’, Dissertation, 2000.
9. Agile Manifesto, http://agilemanifesto.org/
10. Stephen Denning, “The Age of Agile: How Smart Companies Are Transforming the Way Work Gets Done”.
11. Len Bass, Paul Clements, Rick Kazman, “Software Architecture in Practice” 3rd Edition.
12. Antonio Martini, Lars Pareto, Jan Bosch, “Towards introducing Agile Architecting in Large Companies: the CAFFEA framework”.
13. Jan Bosch, “The Three Layer Product Model:An Alternative View on SPLs and Variability”, Keynote, VAMOS 2018, February 2018.
14. Mathias Traub, Alexander Maier, Karl Barbehon, “Future Automotive Architecture and the Impact of IT Trends”, IEEE Software, June 2017.

“Be the change you want to see in the world!” *MAHATMA GANDHI*