

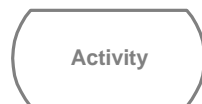
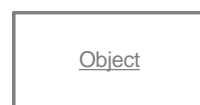
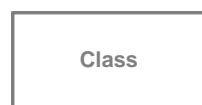
# Extreme Architecture Amplifying the UML

Phil Robinson  
Lonsdale Systems

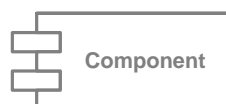
## Unified Modelling Language (UML)



The UML is a notation  
not a methodology



Actor



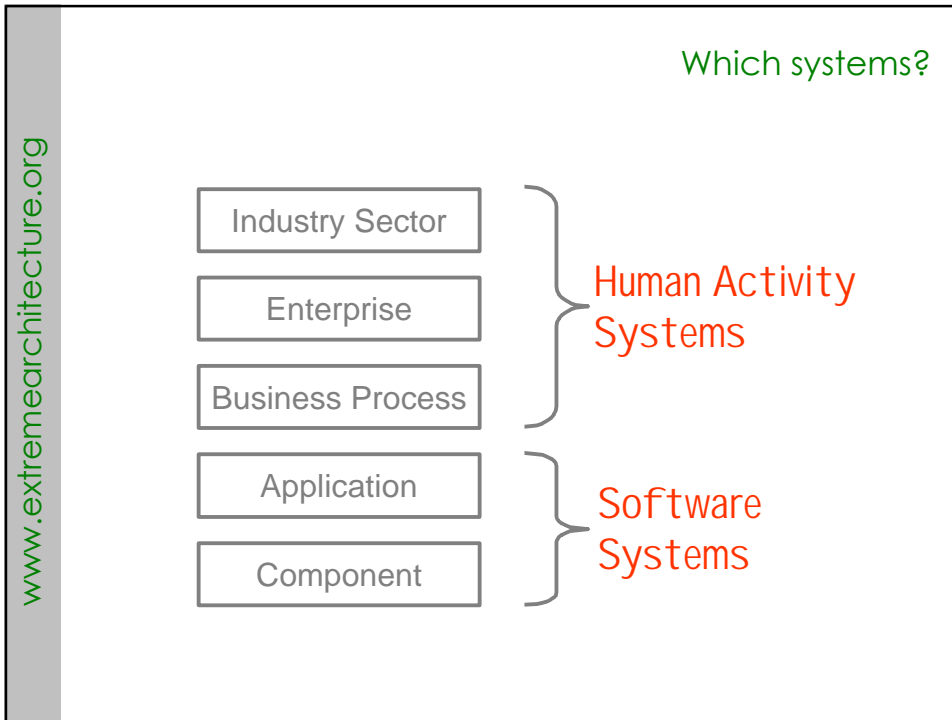
## History of the UML

1995	Unified Method 0.8 (Booch & Rumbaugh)
1996	Unified Modelling Language (UML) 0.9 (Jacobson joins and they become the <i>Three Amigos</i> )
1997	(Jan) - UML 1.0 (Sep) - UML1.1 (adopted by OMG)
1998	UML 1.2
1999	UML 1.3 (first formal version)
2001	UML 1.4
2002	UML 1.4.1
2003	(Mar) - UML 1.5 (Aug) - UML 2.0
????	ISO Standard (based on UML 1.4.1!)

## What is the UML used for?

*The Unified Modeling Language is a visual language for specifying, constructing and documenting the artifacts of systems.*

OMG UML 2.0 Superstructure Specification

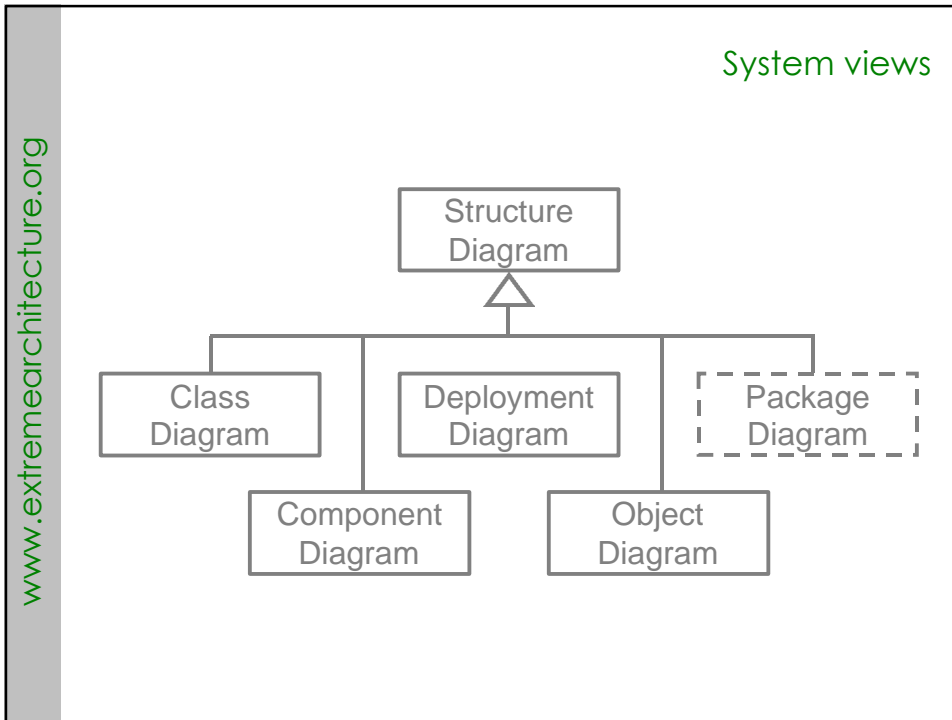


www.extremearchitecture.org

System views

*Structure diagrams show the static structure of the objects in a system.*

OMG UML 2.0 Superstructure Specification

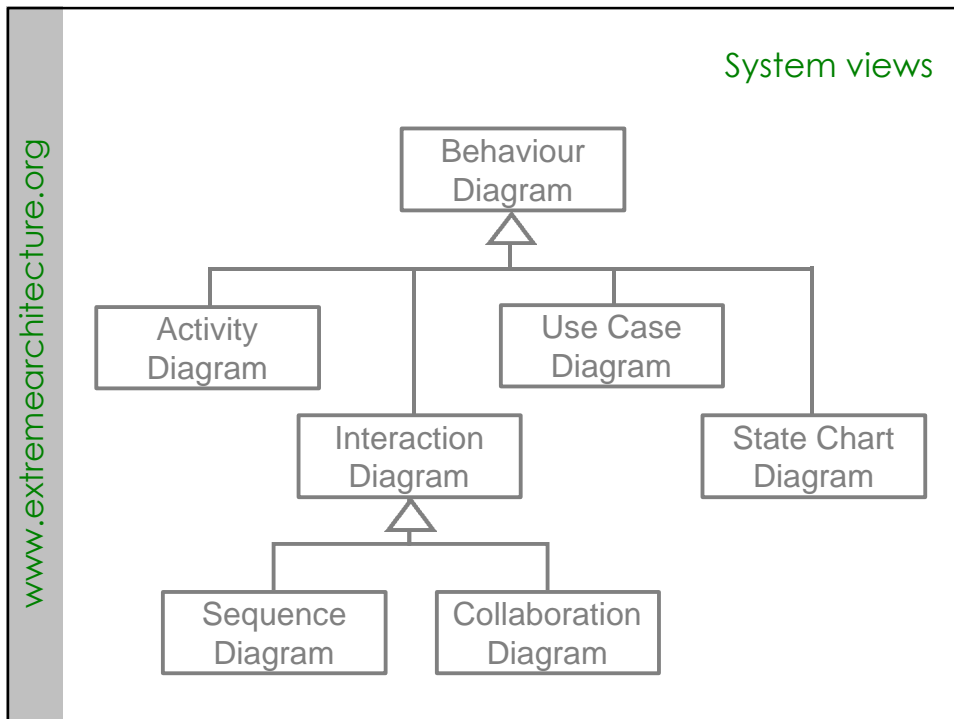


www.extremearchitecture.org

System views

*Behavior diagrams show the dynamic behavior of the objects in a system...*

OMG UML 2.0 Superstructure Specification



- www.extremearchitecture.org
- System views
- The taxonomy of UML diagrams is based on abstract system views
  - This has two disadvantages
    - Diagrams tend to be drawn because they are “in the UML and therefore must be required”
    - There is an assumption that these are the only types of diagram that can be drawn

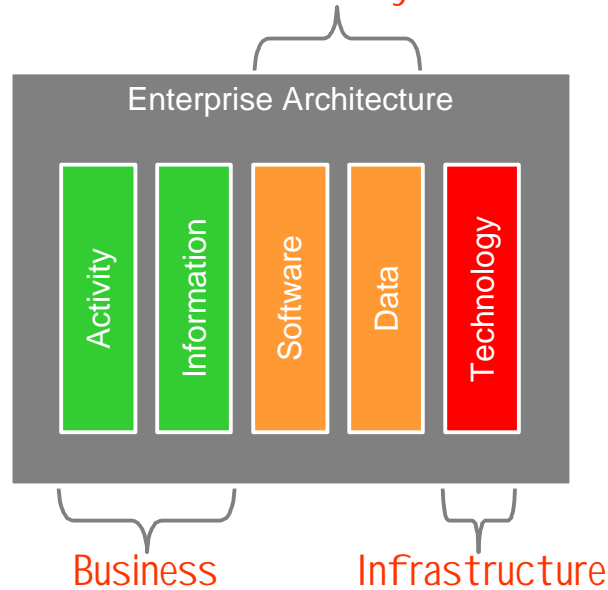
## System views

*...this taxonomy provides a logical organization for the various major kinds of diagrams. However, it does not preclude the mixing different kinds of diagram types...*

*...the boundaries between the various kinds of diagram types are not strictly enforced...*

OMG UML 2.0 Superstructure Specification

## The enterprise architecture view Information Systems



## The Extreme Architecture framework

www.extremearchitecture.org

	Activity	Information	Software	Data	Technology
Sector	Architecture Content				
Enterprise					
Process					
Application					
Component					

## Why extreme?

www.extremearchitecture.org

- Pragmatic – adopts a ‘middle path’ between the extremes of chaos and perfection
- Easy to describe – single 6 x 6 matrix
- Minimal work products – simple bullet point lists will do!
- Unifies disparate disciplines
  - Business modelling
  - Requirements analysis
  - Software construction
  - IT infrastructure management
- Encourages a shared understanding

Business modelling

www.extremearchitecture.org

	Activity	Information	Software	Data	Technology
Sector					
Enterprise					
Process					
Application					
Component					

Business modelling

www.extremearchitecture.org

	Activity	Information	Software	Data	Technology
Sector	Scope	Subject Areas			
Enterprise	Activities	Information Requirements	Functional Areas	Business Objects	
Process	Workflow				
Application					
Component					



Requirements analysis

www.extremearchitecture.org

	Activity	Information	Software	Data	Technology
Sector	Scope Activities Workflow	Subject Areas Information Requirements	Functional Areas	Business Objects	
Enterprise					
Process					
Application	Use Cases	Interface Requirements	Vision Functional Requirements Non-Functional Requirements	Storage Requirements	
Component					

Software construction

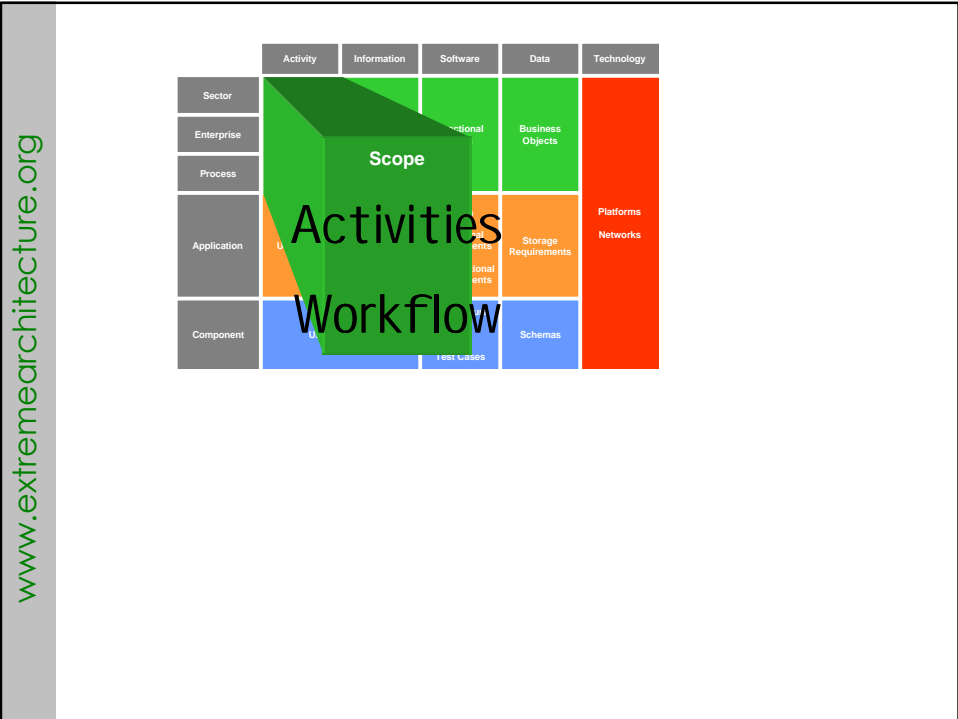
www.extremearchitecture.org

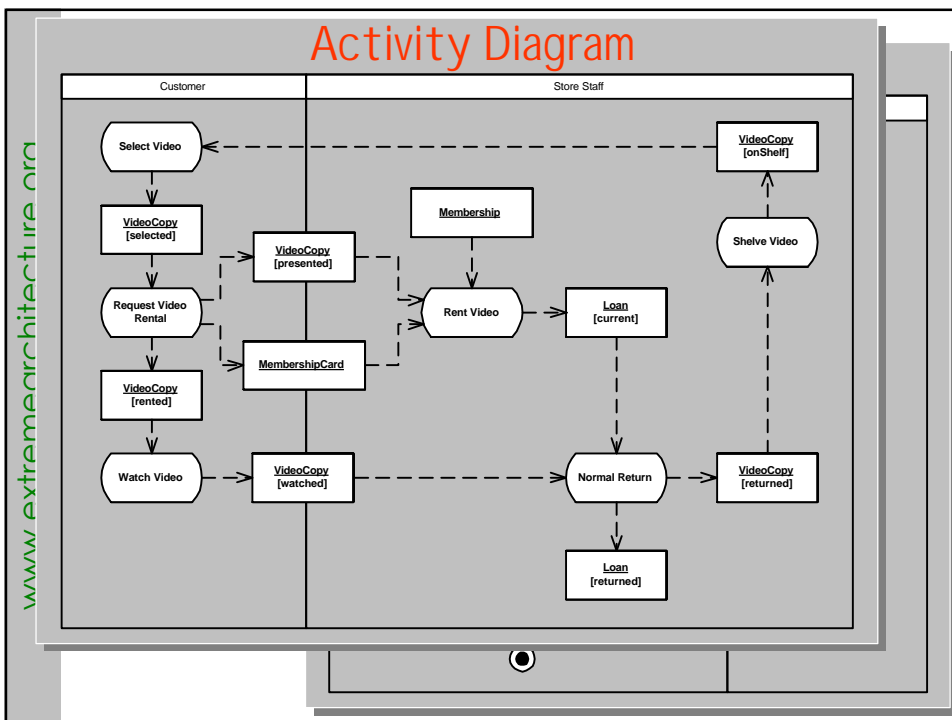
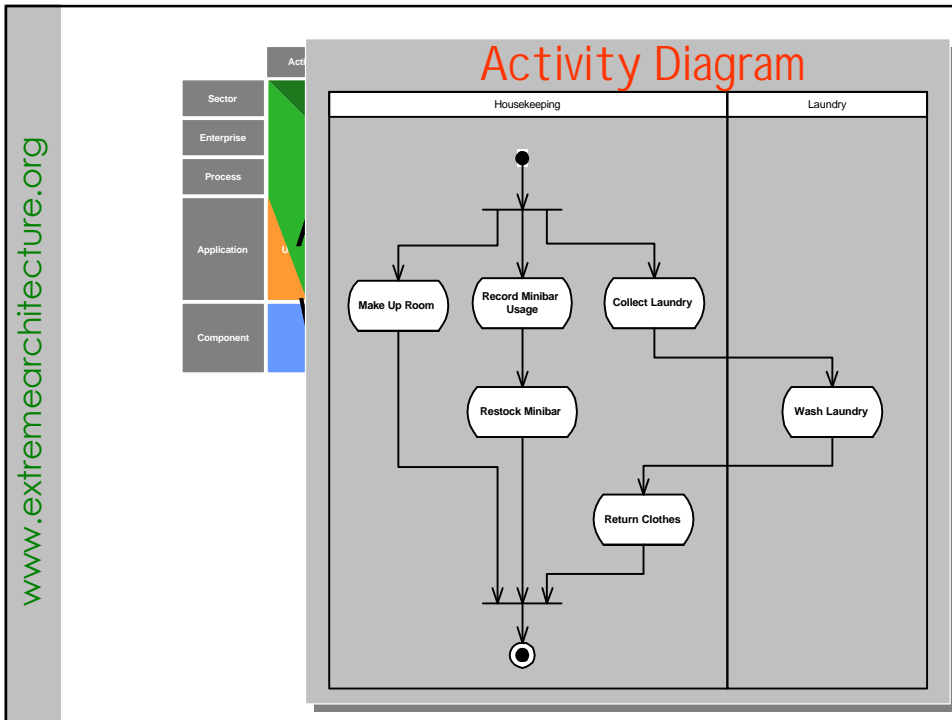
	Activity	Information	Software	Data	Technology
Sector	Scope Activities Workflow	Subject Areas Information Requirements	Functional Areas	Business Objects	
Enterprise					
Process					
Application	Use Cases	Interface Requirements	Vision Functional Requirements Non-Functional Requirements	Storage Requirements	
Component	User Interface		Architecture Code Test Cases	Schemas	

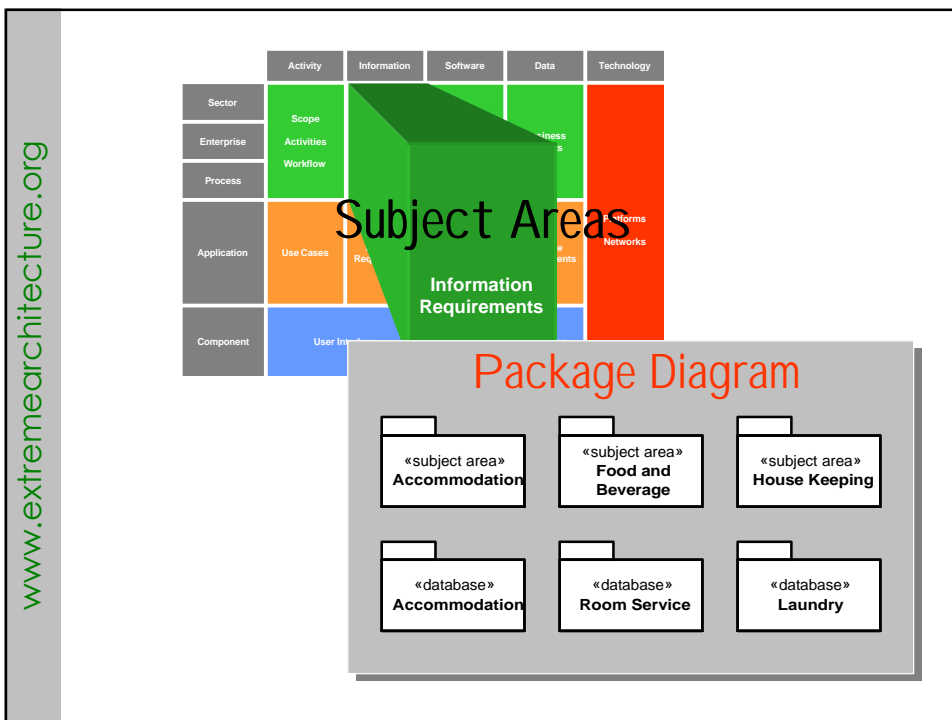
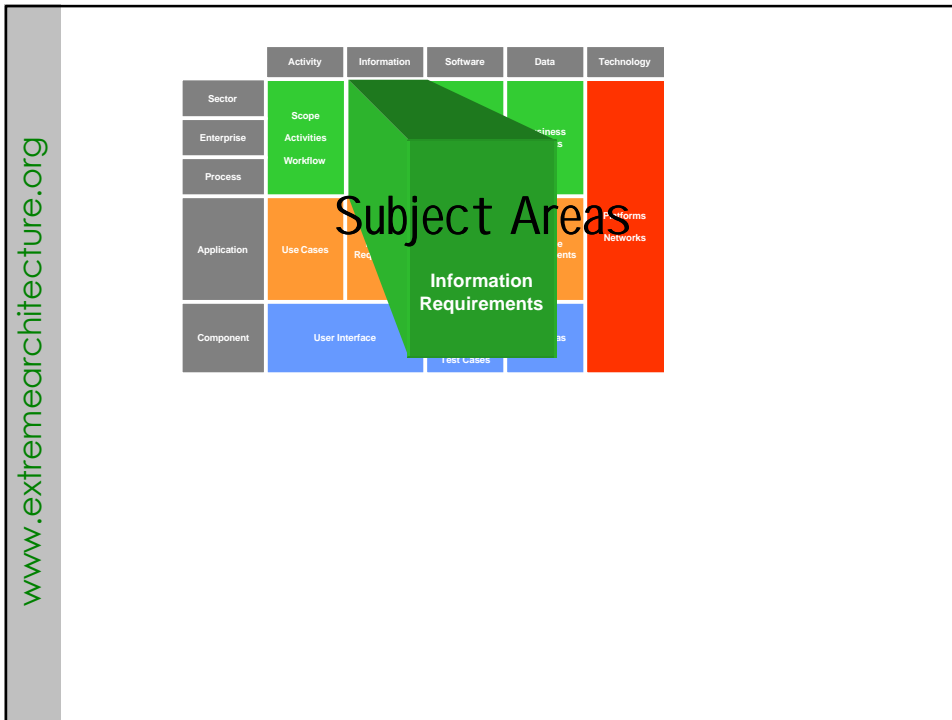
Infrastructure management

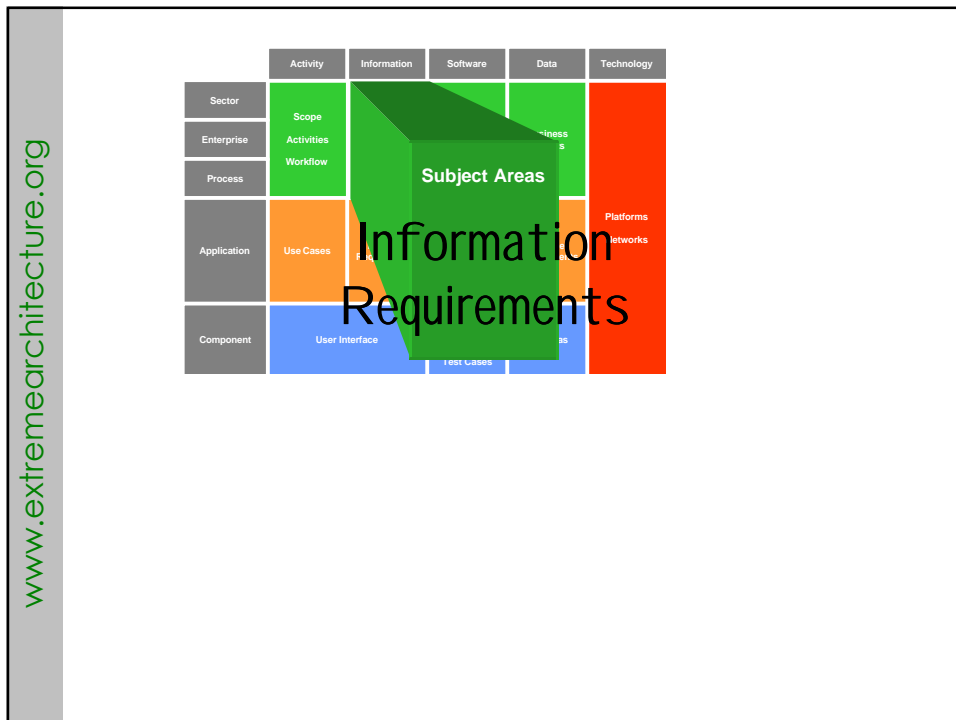
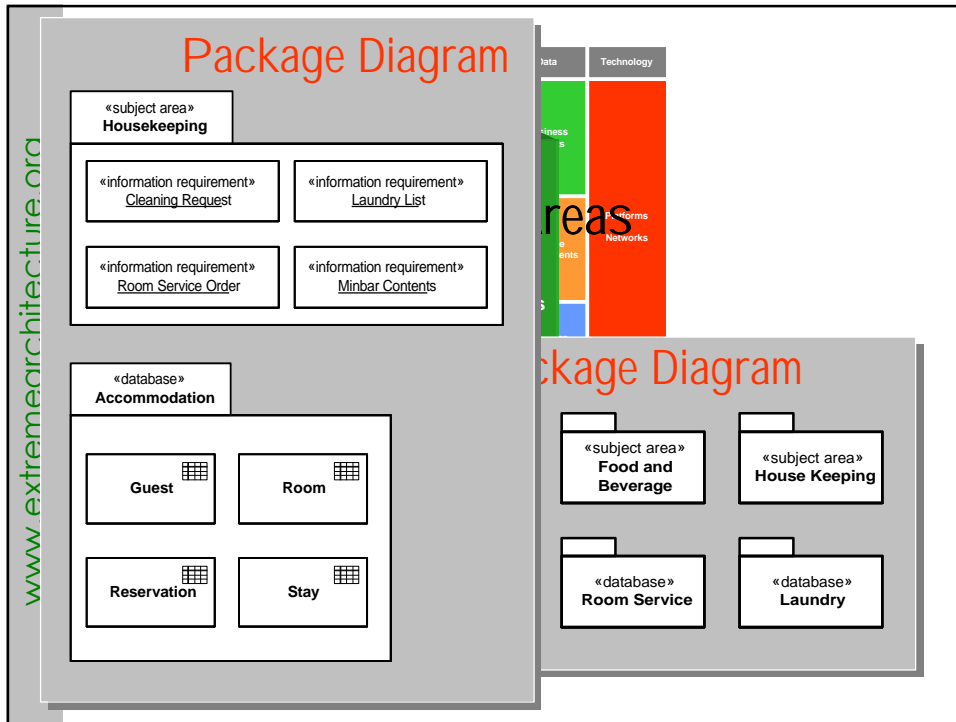
www.extremearchitecture.org

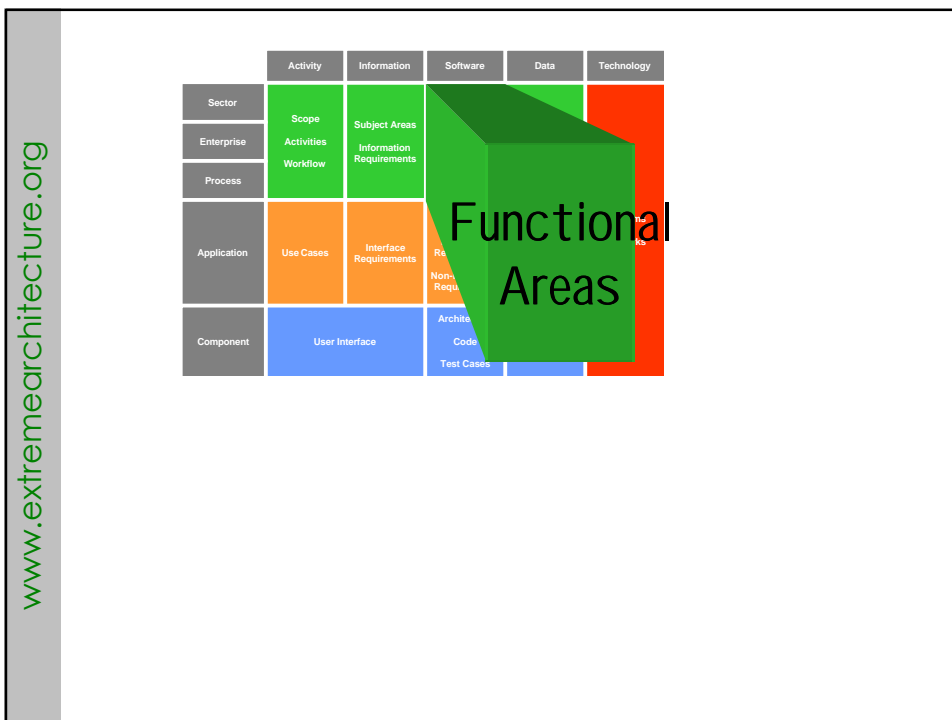
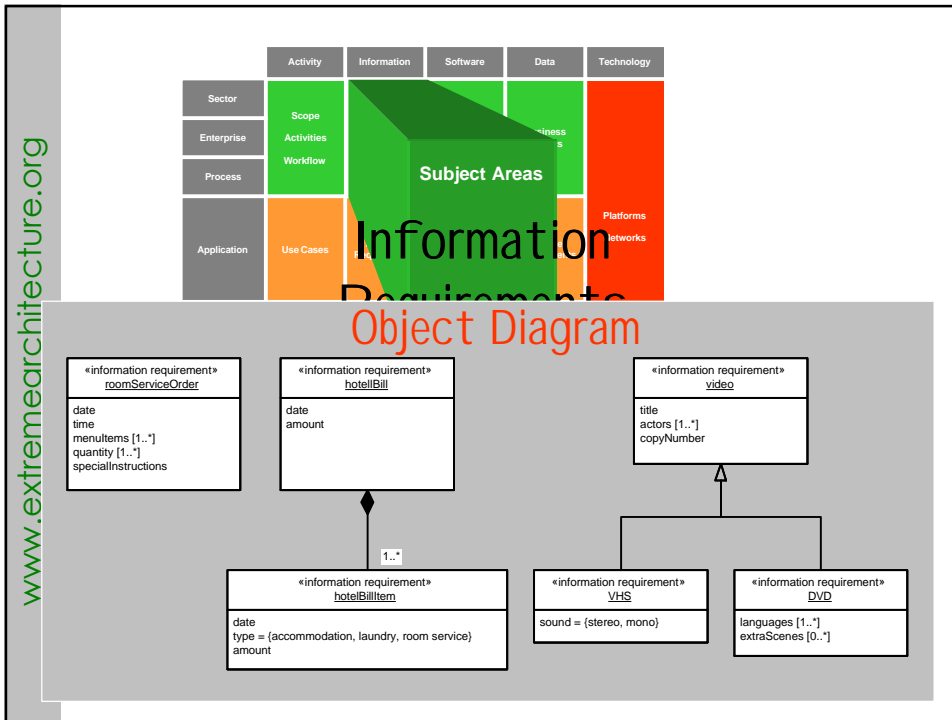
	Activity	Information	Software	Data	Technology
Sector	Scope Activities Workflow	Subject Areas Information Requirements	Functional Areas	Business Objects	Platforms Networks
Enterprise					
Process					
Application	Use Cases	Interface Requirements	Vision Functional Requirements Non-Functional Requirements	Storage Requirements	
Component			User Interface		

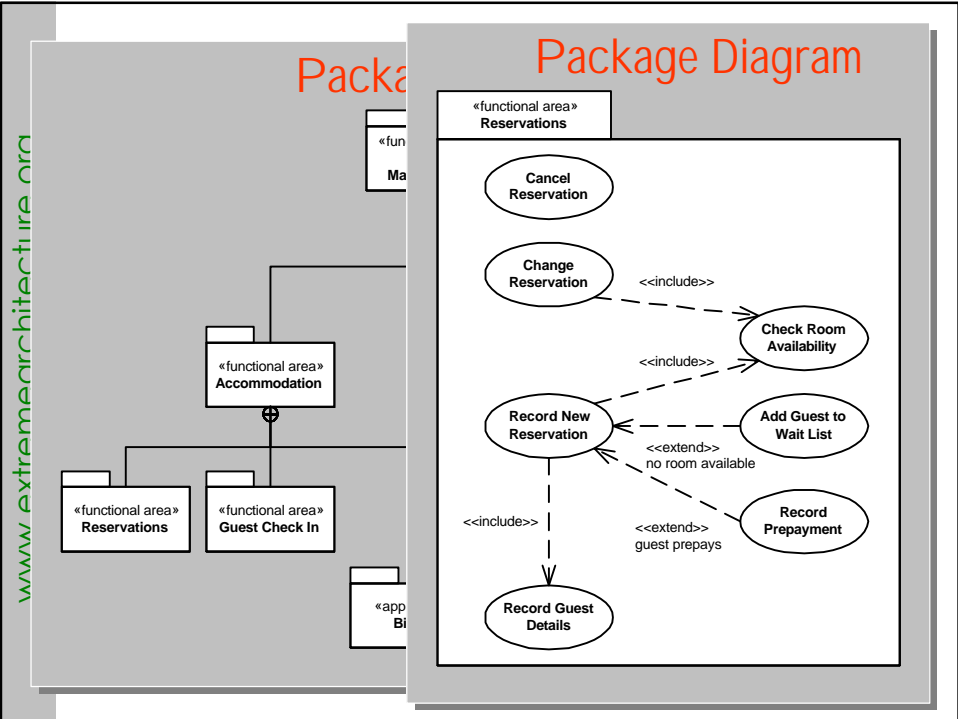
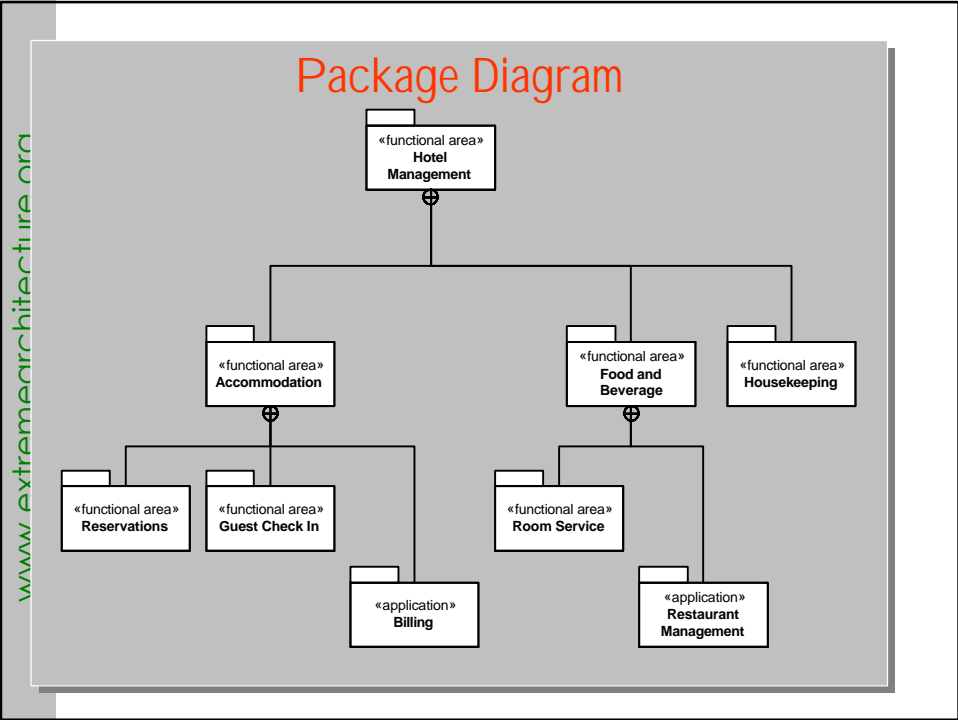


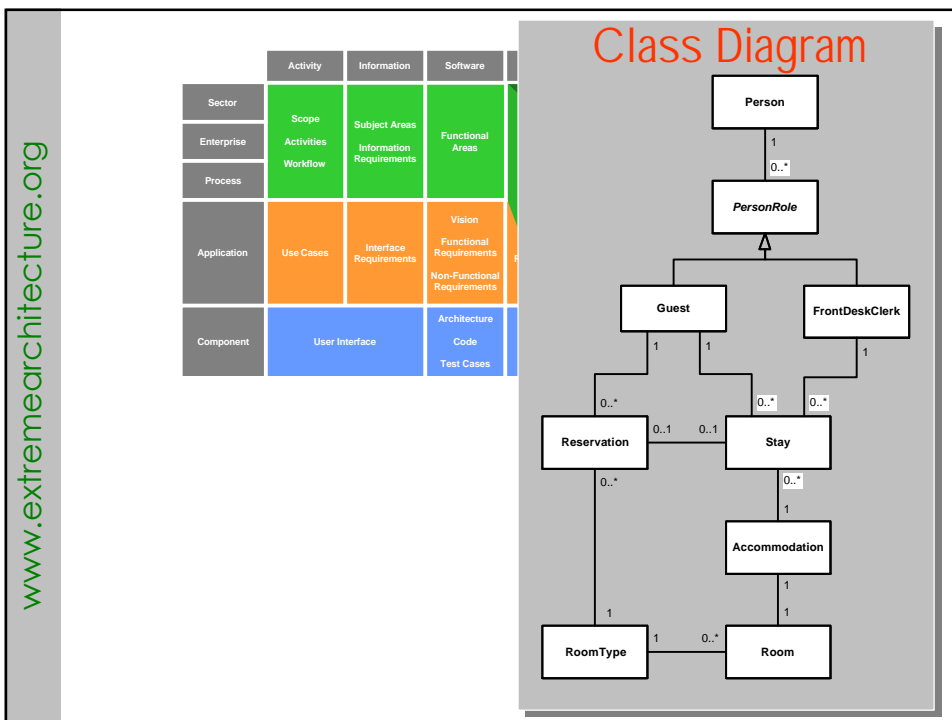




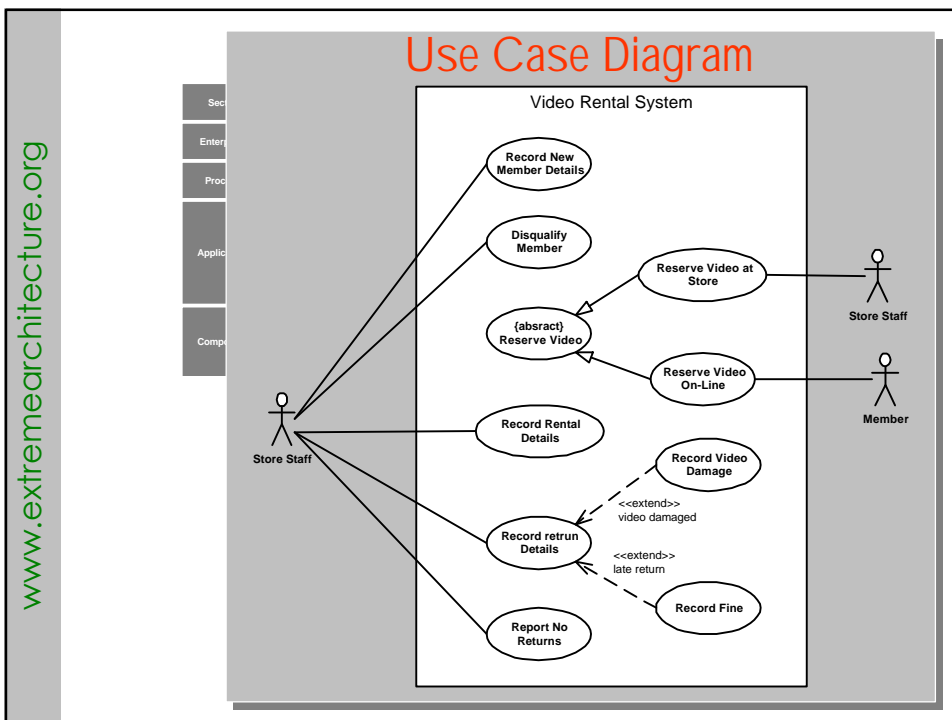
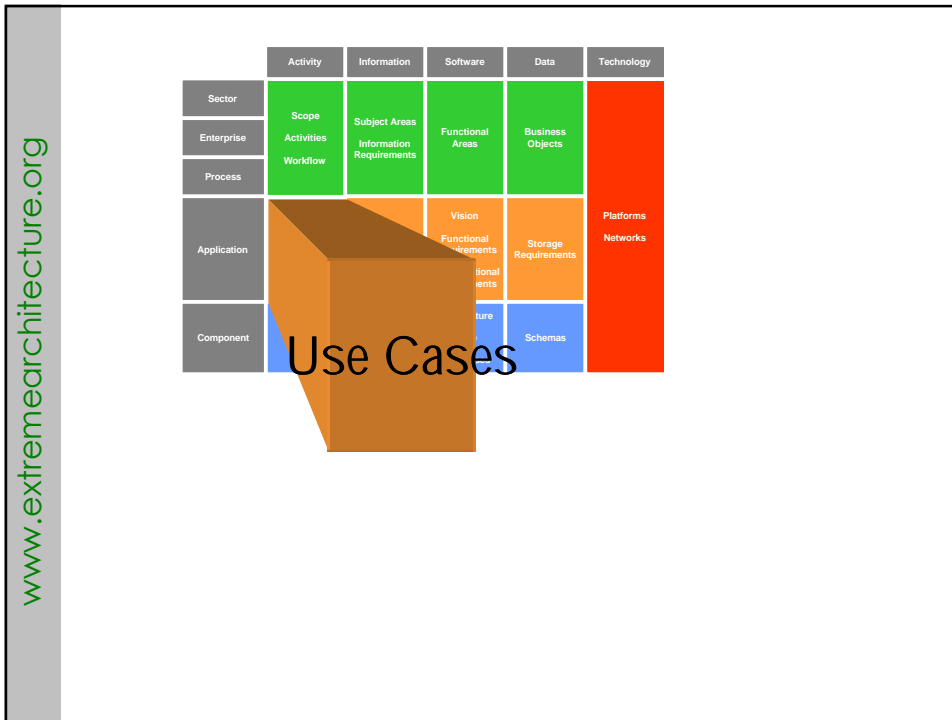


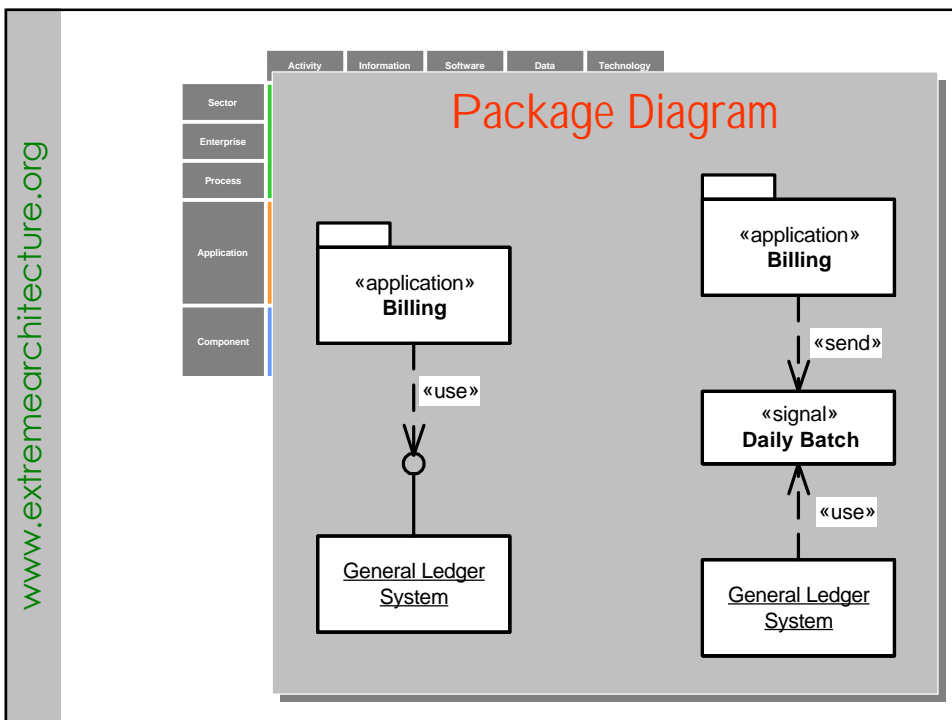
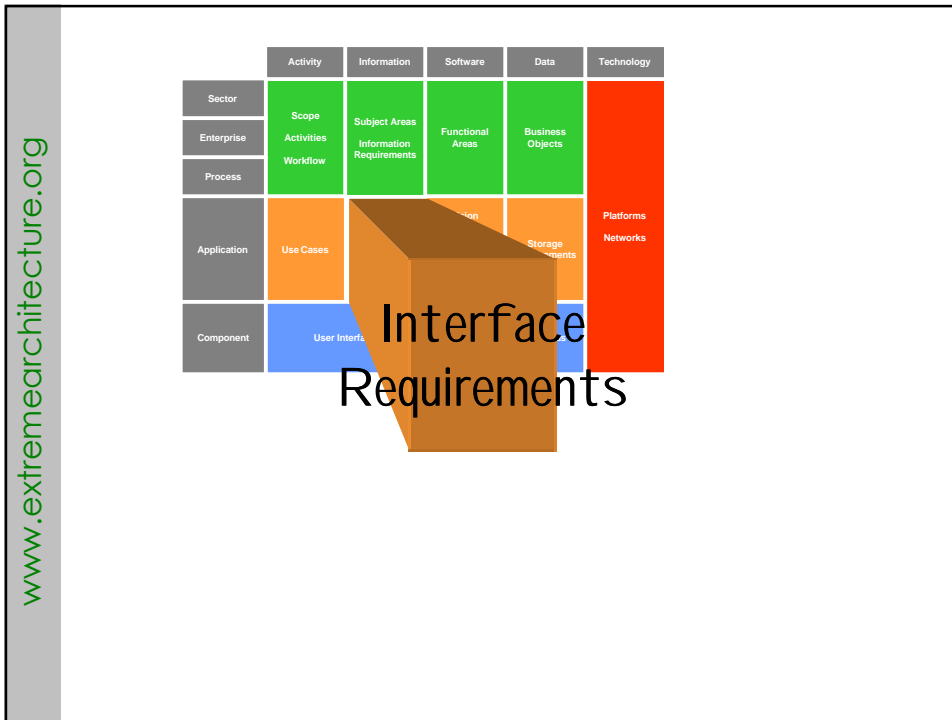


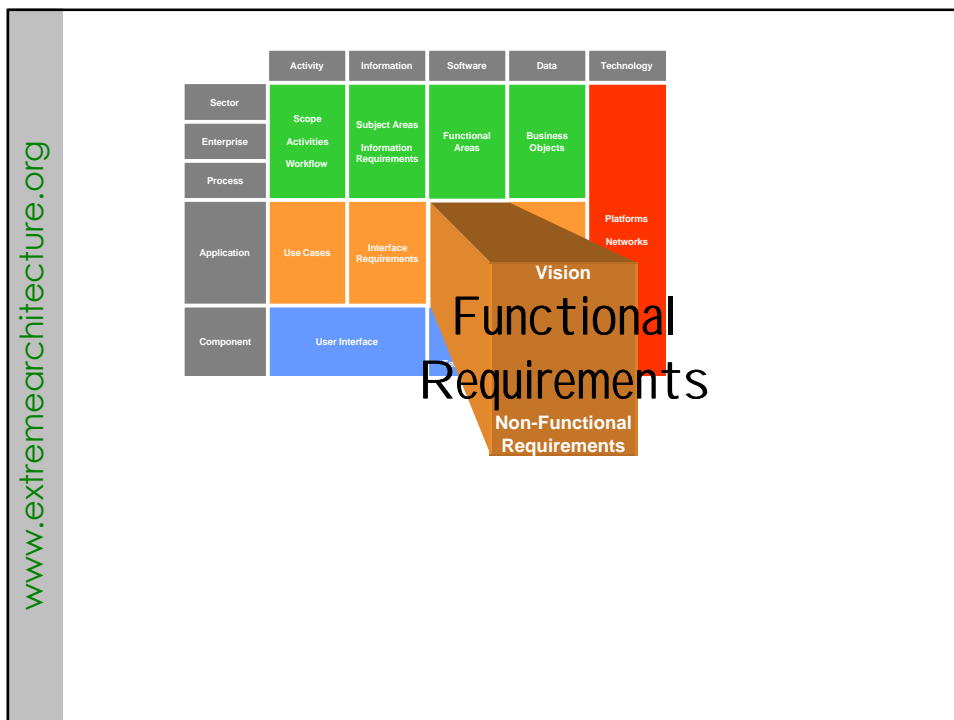
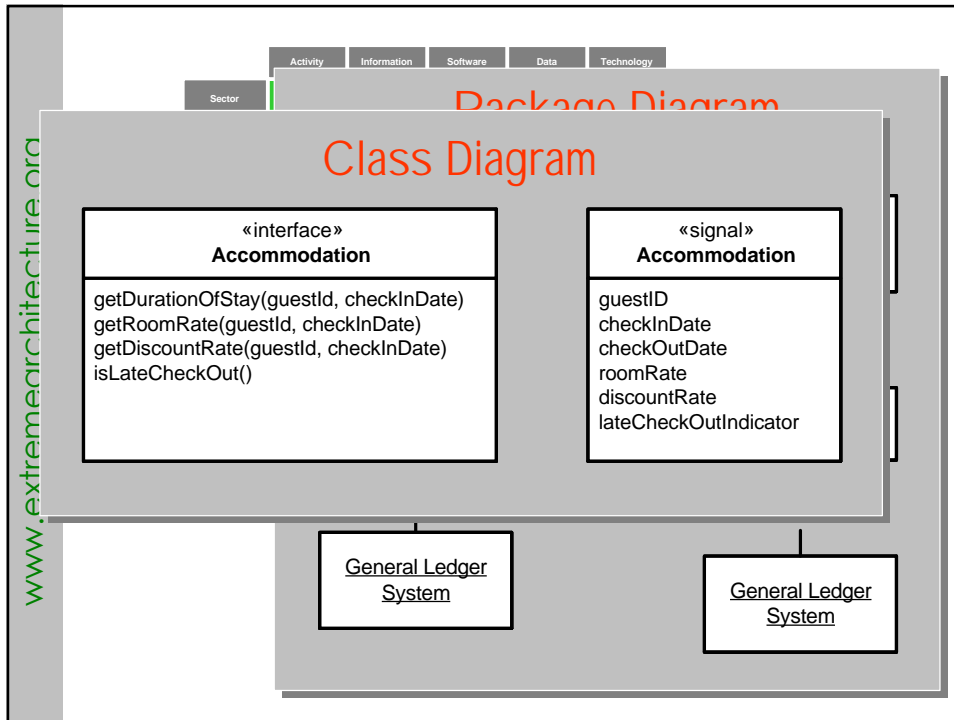


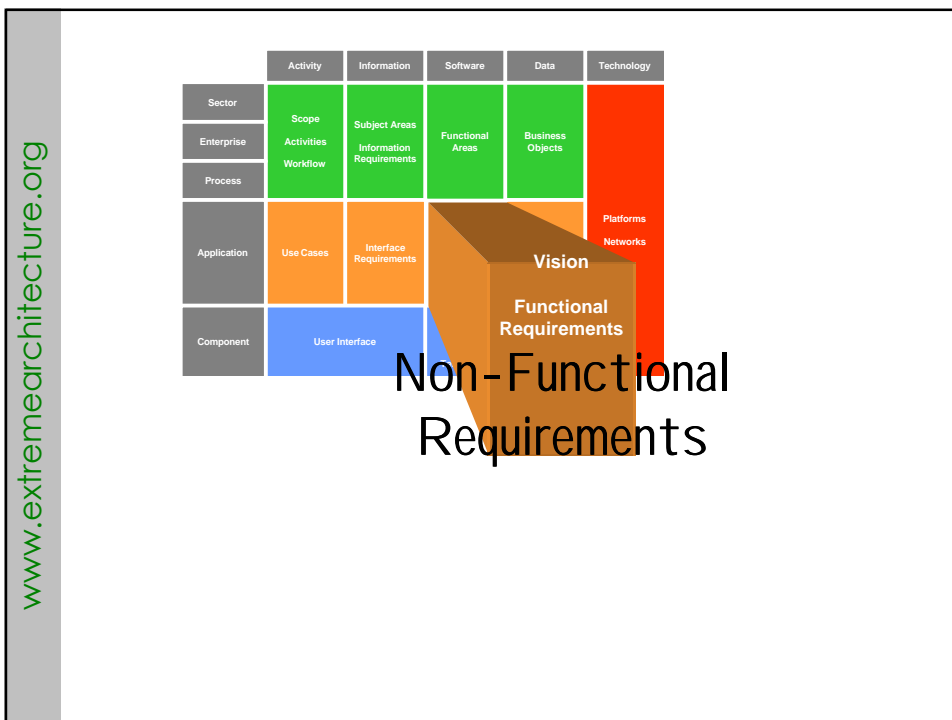
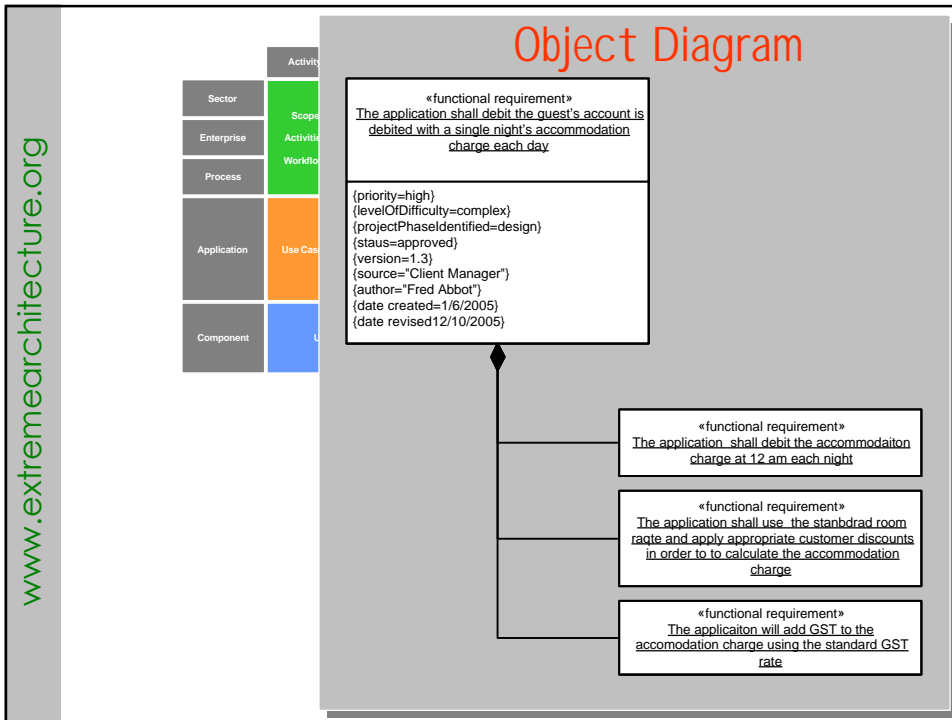


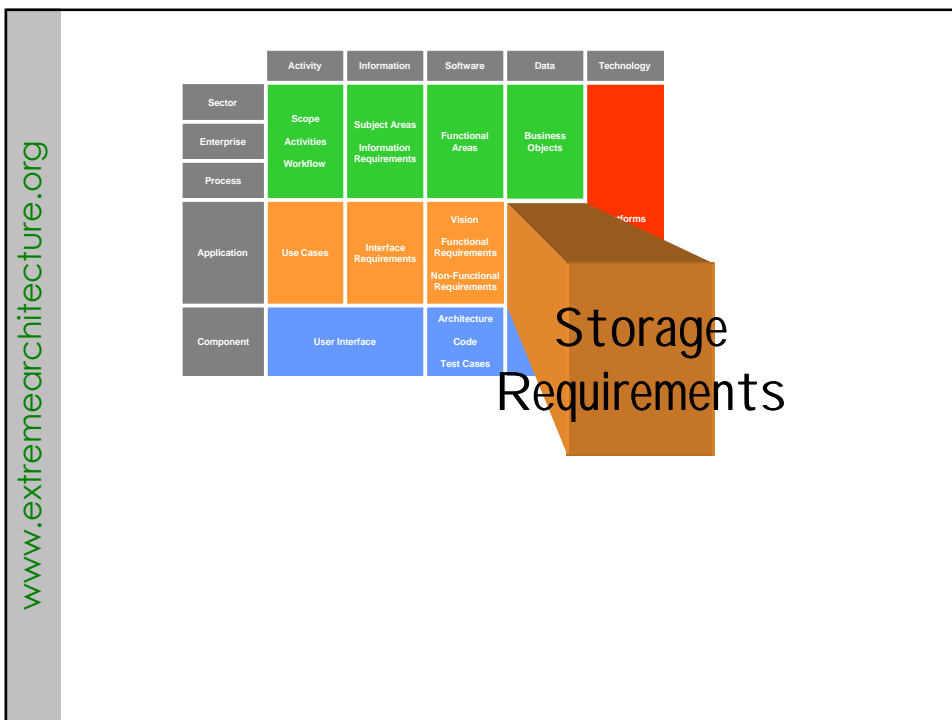
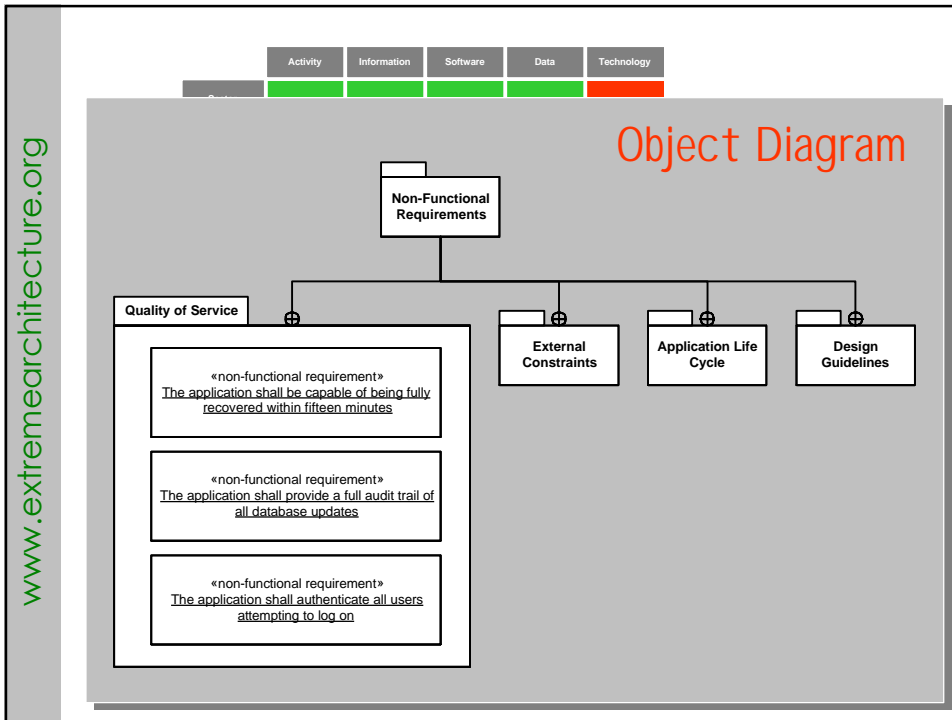


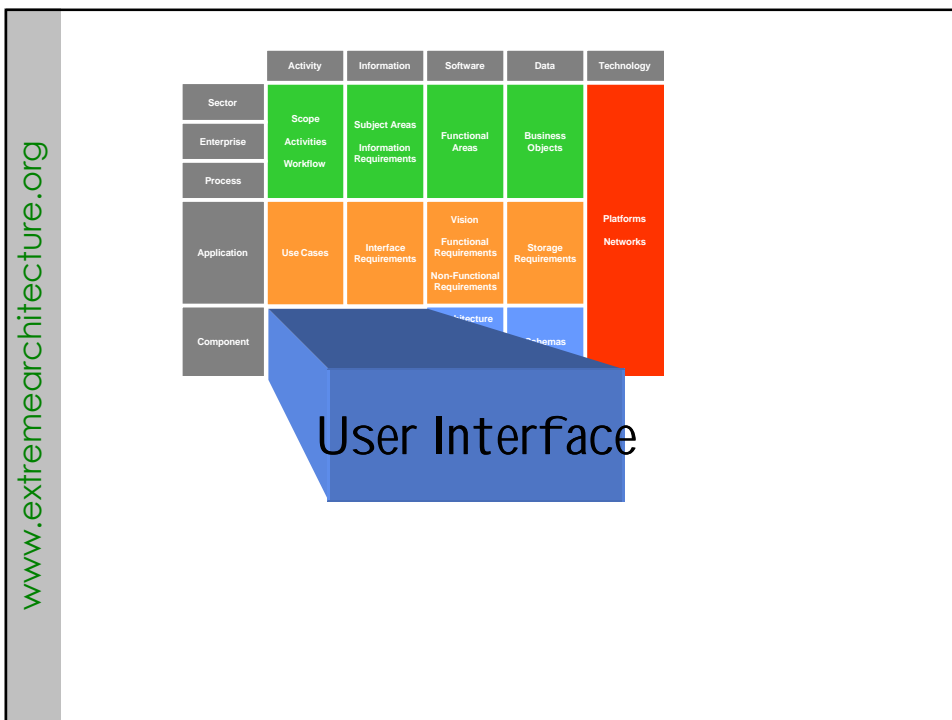
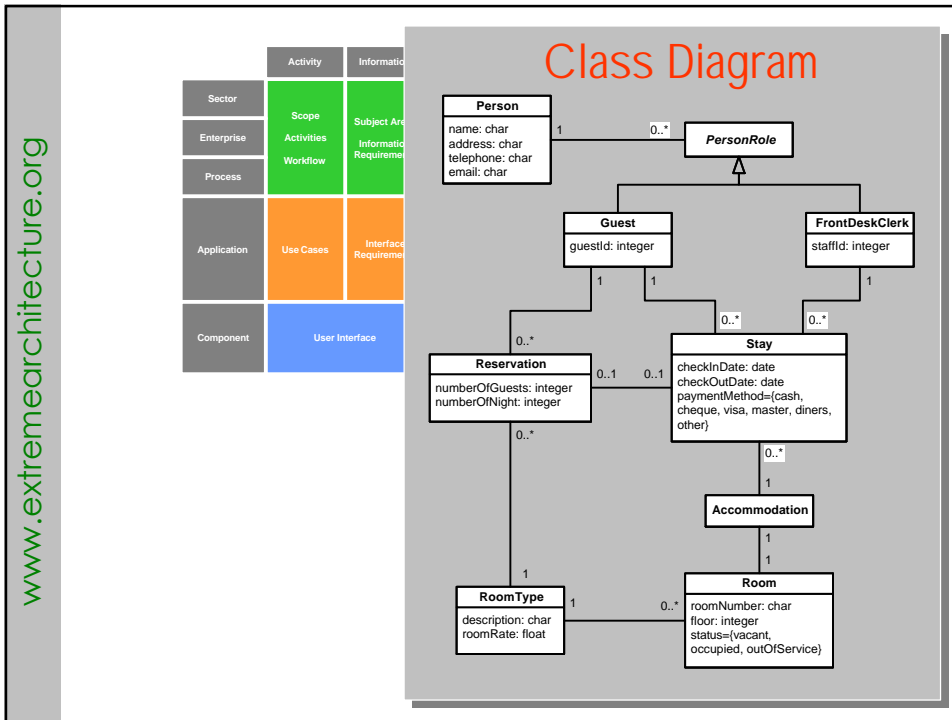


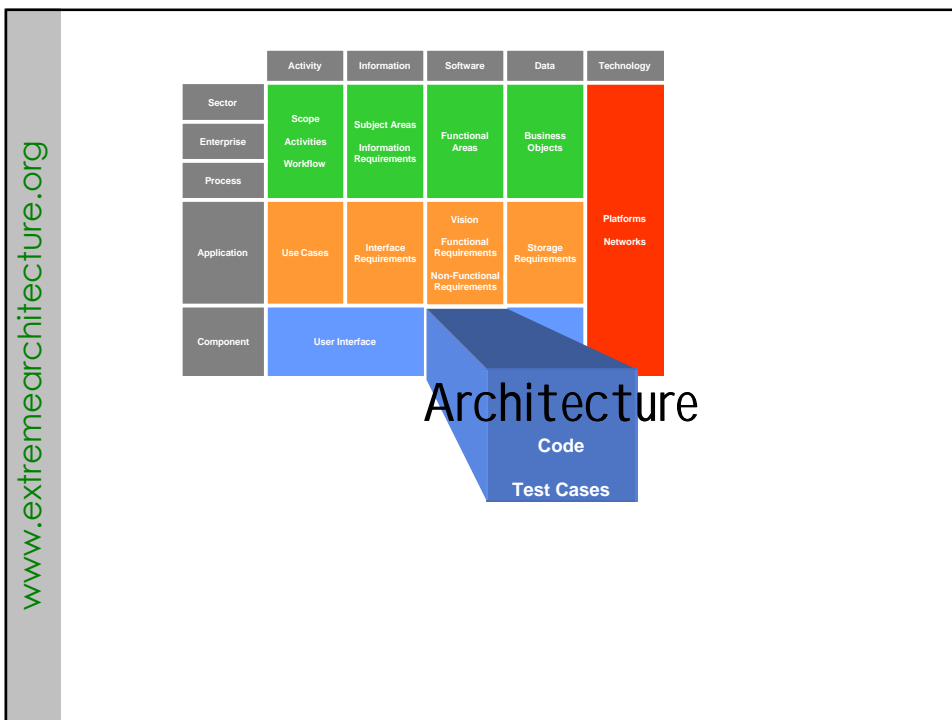
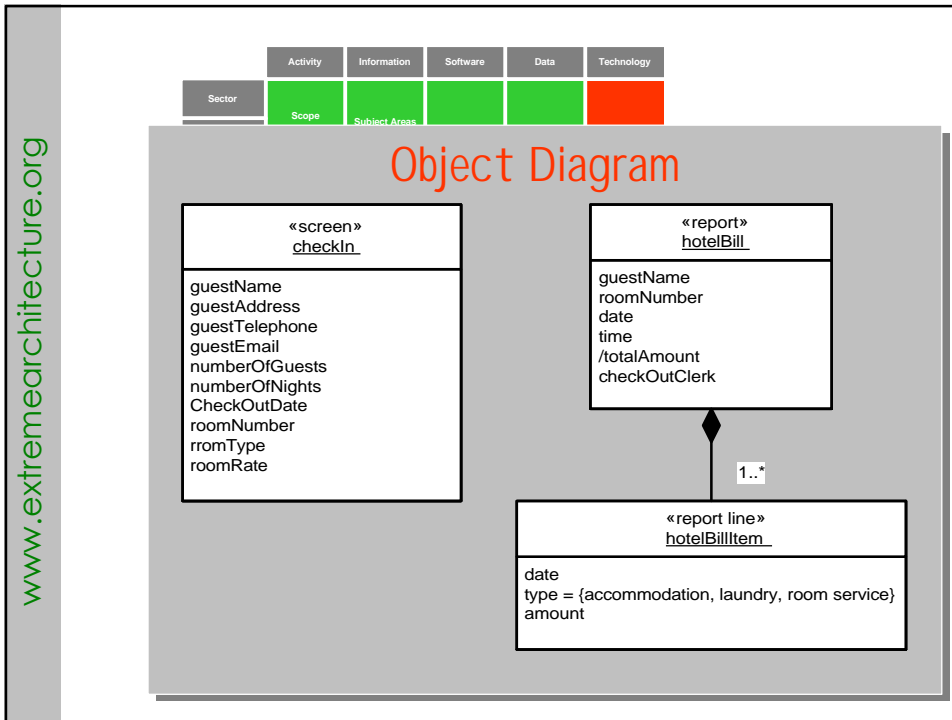


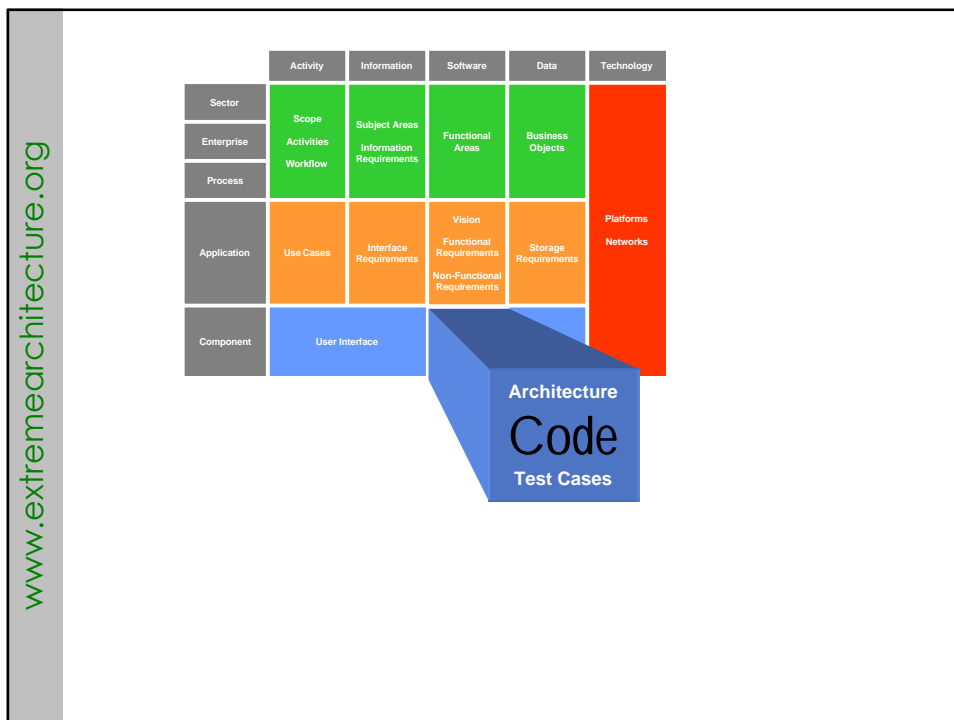
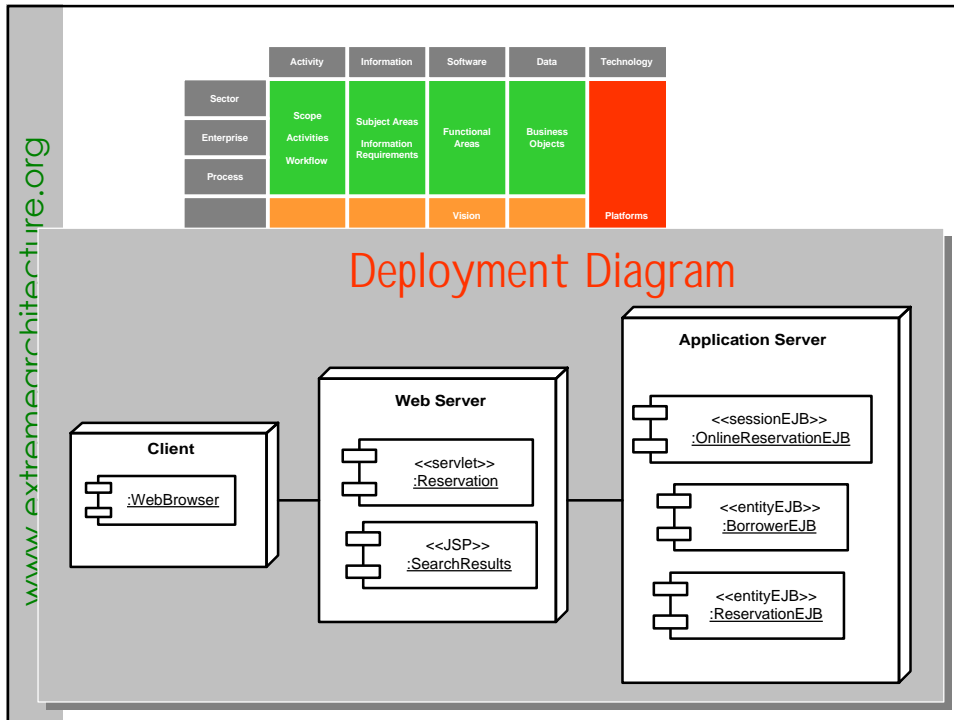




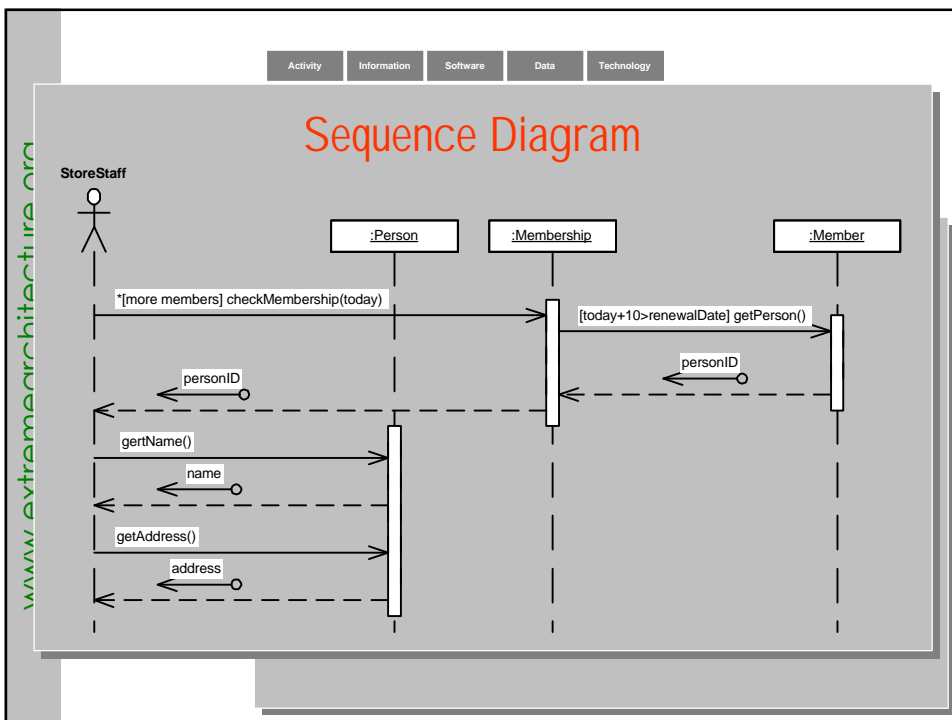
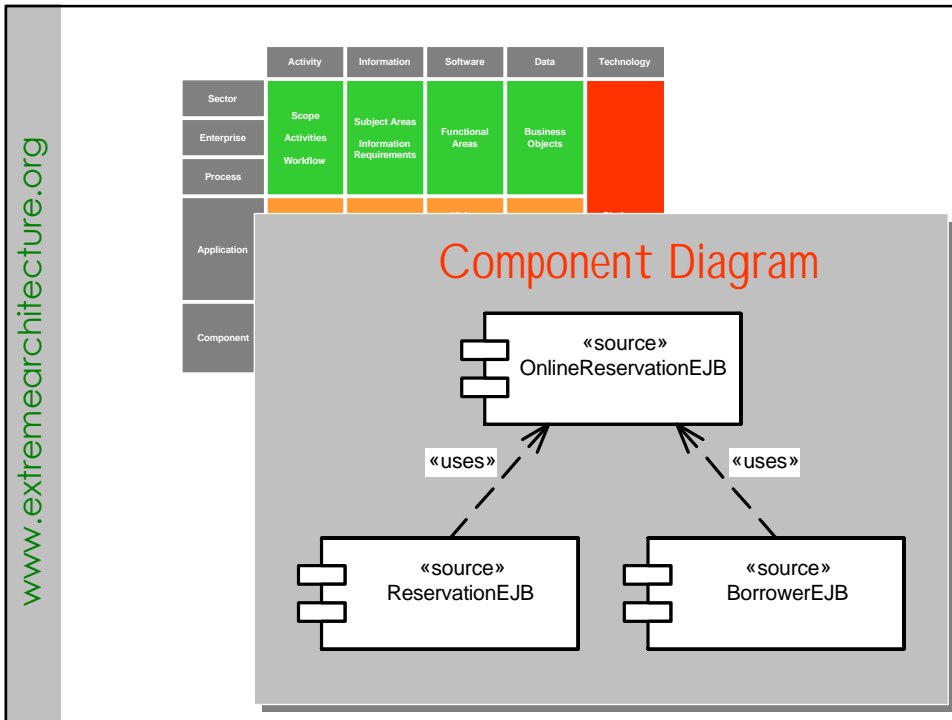


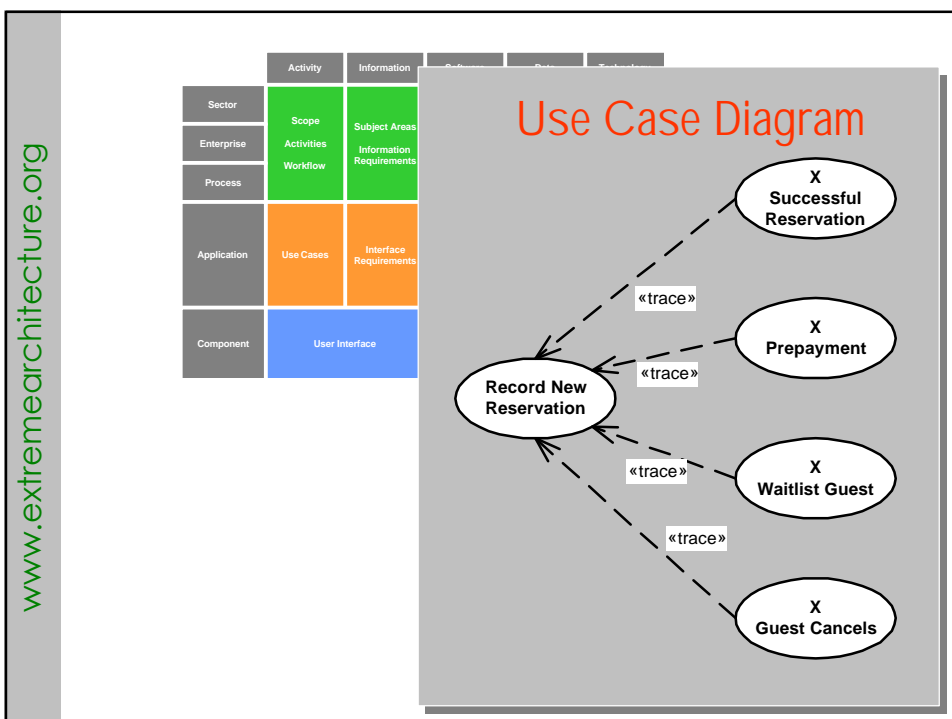
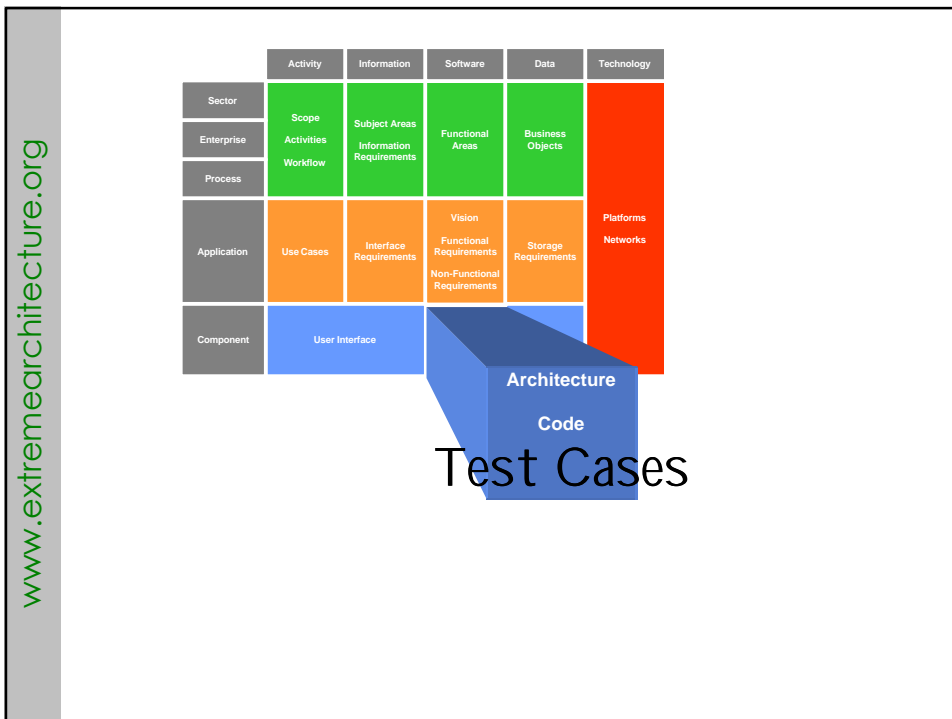


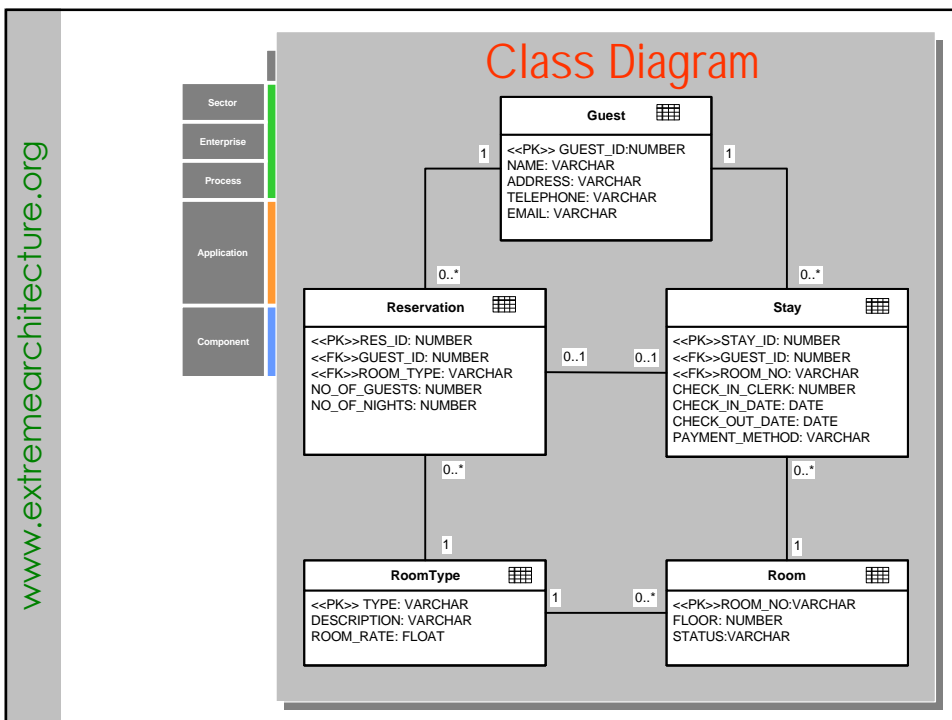
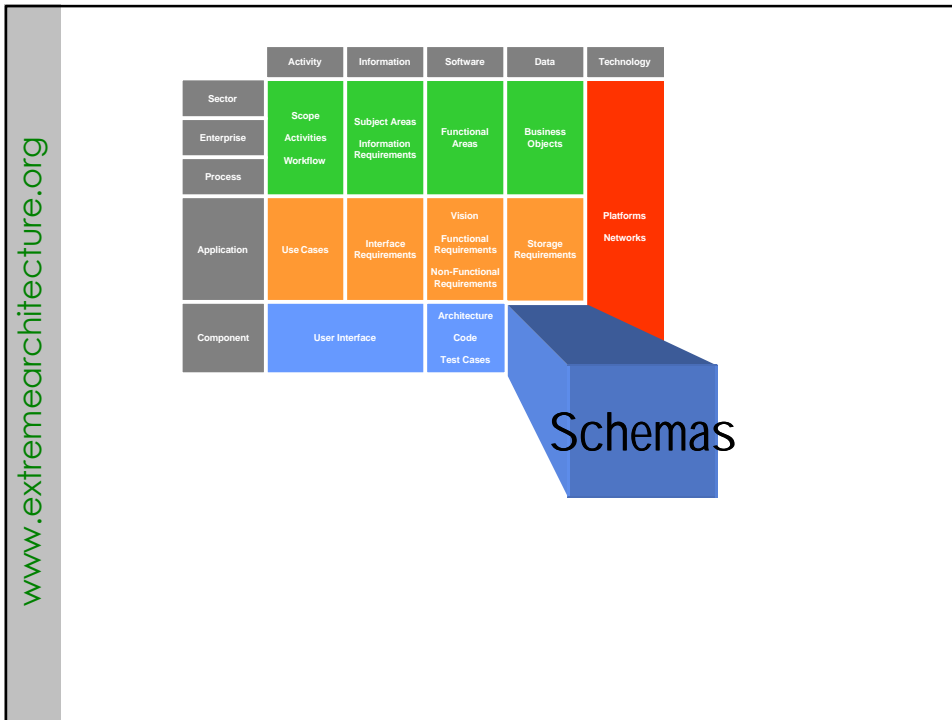


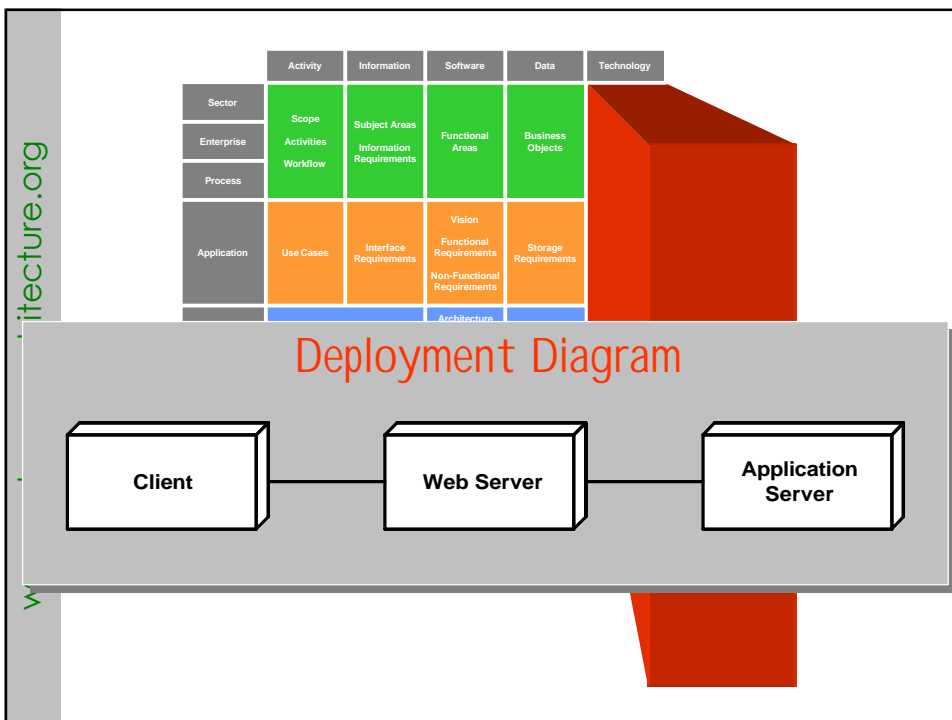
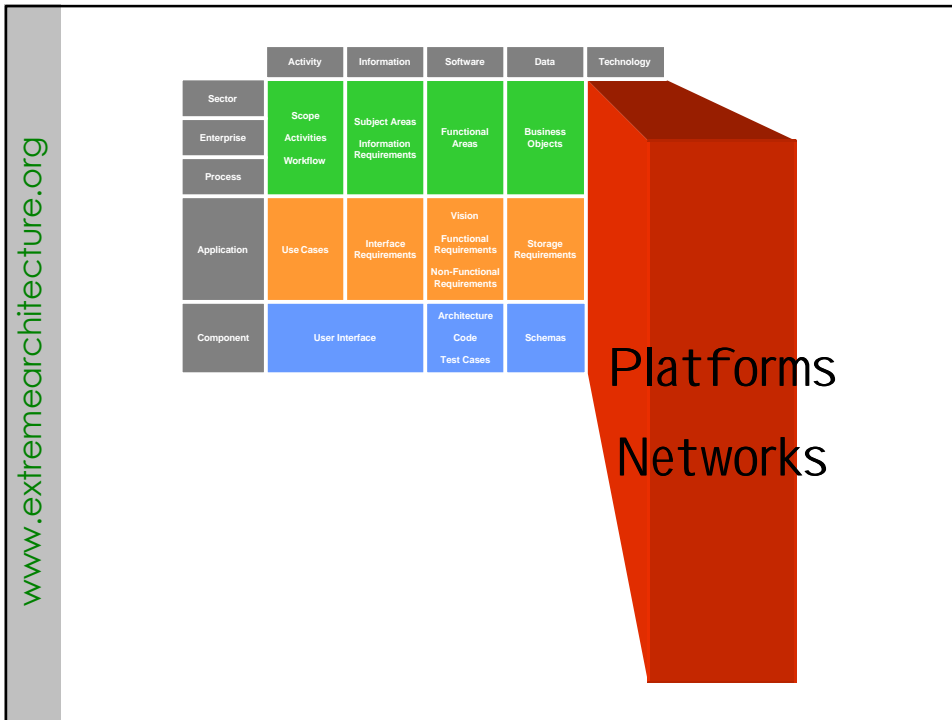












## But Remember minimal work products

	Activity	Information	Software	Data	Technology
Sector	<ul style="list-style-type: none"> <li>Promotion of new releases</li> </ul>				
Enterprise	<ul style="list-style-type: none"> <li>Membership</li> <li>Video library</li> <li>Reservations</li> <li>Rentals</li> <li>Returns</li> </ul>	<ul style="list-style-type: none"> <li>New release details</li> <li>Member details</li> <li>Video details</li> <li>Reservation details</li> <li>Rental details</li> </ul>	<ul style="list-style-type: none"> <li>General ledger</li> <li>Accounts receivable</li> <li>Asset register</li> <li>Casual staff payroll</li> <li>Rental management</li> </ul>	<ul style="list-style-type: none"> <li>Member</li> <li>Video title</li> <li>Video copy</li> <li>Rental</li> </ul>	
Process	<ul style="list-style-type: none"> <li>Returns                             <ul style="list-style-type: none"> <li>Normal Return</li> <li>Late Return</li> <li>Follow up overdue videos</li> </ul> </li> <li>...</li> </ul>				<ul style="list-style-type: none"> <li>Intel server</li> <li>ADSL internet connection</li> </ul>
Application	<ul style="list-style-type: none"> <li>Record return</li> <li>Record fine paid</li> </ul>	<ul style="list-style-type: none"> <li>Rental details                             <ul style="list-style-type: none"> <li>Date out</li> <li>Date due back</li> <li>Actual return date</li> <li>Fine due</li> <li>Fine paid</li> <li>...</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Notify overdue loans</li> <li>Calculate fine due</li> <li>Read video and membership card bar codes</li> </ul>	<ul style="list-style-type: none"> <li>Rental                             <ul style="list-style-type: none"> <li>Date out</li> <li>Actual return date</li> <li>Fine paid</li> <li>...</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Linux</li> <li>Apache web server</li> <li>MYSQL</li> <li>PHP</li> </ul>
Component	<ul style="list-style-type: none"> <li>Member details screen</li> <li>Return screen</li> <li>Fine window</li> <li>Overdue report</li> </ul>		<p><i>These are the deliverables of the software project that builds the application!</i></p>		

## Questions

Phil Robinson  
Lonsdale Systems