



Endorsed Education Provider



BUSINESS ANALYST WORKSHOP

COURSE BOOK



IT & Business Analysis | training | placement | consulting

RequirementsInc.com | 703 468 1921



/RequirementsInc



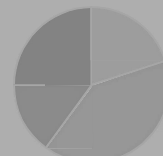
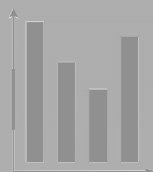
/RequirementsInc



1 IT PROJECTS & BUSINESS ANALYSIS - INTRODUCTION

1.1 IT Project and Key Players

IDENTIFY DIFFERENT TEAMS THAT A BA INTERACTS WITH
UNDERSTAND CORE BA RESPONSIBILITIES



What is a Project?

- Temporary, time-bound endeavor
- Has a certain goal
- Gets something done to improve the abilities / efficiencies of someone or an organization
- Performed with **limited time, money and resources** - efficiency is key

Example

Building a house, Going on a trip, Implementing a time tracking system as a replacement to paper time cards

FYI

In this course, the terms Software, System, Software System and Project are used interchangeably to refer to an IT Project aimed at solving a business problem.



Business Analyst Workshop



Endorsed Education Provider
Endorsed Program™

3

Project Participants

- **Business Sponsor** – Visionary for a software project who can articulate business benefits and justification (Objectives), helps define the boundary (Scope)
- **Business Architect** – Senior BA who ensures business objectives of a project align with the long term business goals
- **Solutions Architect** – Senior developer who leads the design of systems that address business problems
- **Developer/Programmer** – Writes computer programs to automate a business process



Business Analyst Workshop



Endorsed Education Provider
Endorsed Program™

4

Project Participants

- **Tester/QC** – Verifies that the system matches requirements
- **Business Users** – Provide input and feedback for the software being developed
- **Subject Matter Experts (SMEs)** – Have extensive knowledge about existing business processes, help with key decisions



Project Participants

Business Analyst

- Liaison between the users of a software system and the technology team that creates or modifies the software (developers, testers)
- Elicits (extracts) the needs of the users
 - Understand how business users currently perform their tasks (as is) and how the software system (to be) must solve the business problems/challenges
- Define requirements – “what” the system must do
- Ensure that the software solves the business problem

FYI

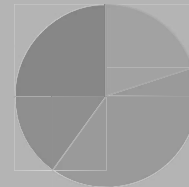
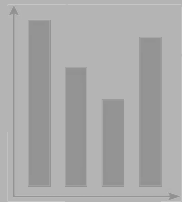
A popular acronym for well defined requirements: Specific, Measurable, Achievable, Realistic, Traceable (SMART).



1.2 Software Development Lifecycle

LEARN TYPICAL STEPS IN PROJECT EXECUTION

UNDERSTAND CONTRASTING METHODS



SDLC: Process by which complex software is created through a systematic approach

The 6 primary Software development activities are

1 Planning

(Gathering requirements for the software and analyzing the scope of the development.)



2

Implementation

(The part of the process where software engineers actually program the code.)



3

Documenting

(This is done for the purpose of future maintenance and enhancement throughout development.)



4

Testing

(An integral and supremely important software development phase, this process ensures that defects are recognized a.s.a.p. and the resulting software is reasonably bug-free.)



5

Deployment

(After the software has been tested and approved for release, deployment might entail installation, customization (like setting parameters for the customer after installation), training on using the software, and there is sometimes an extended period of evaluation.)



6

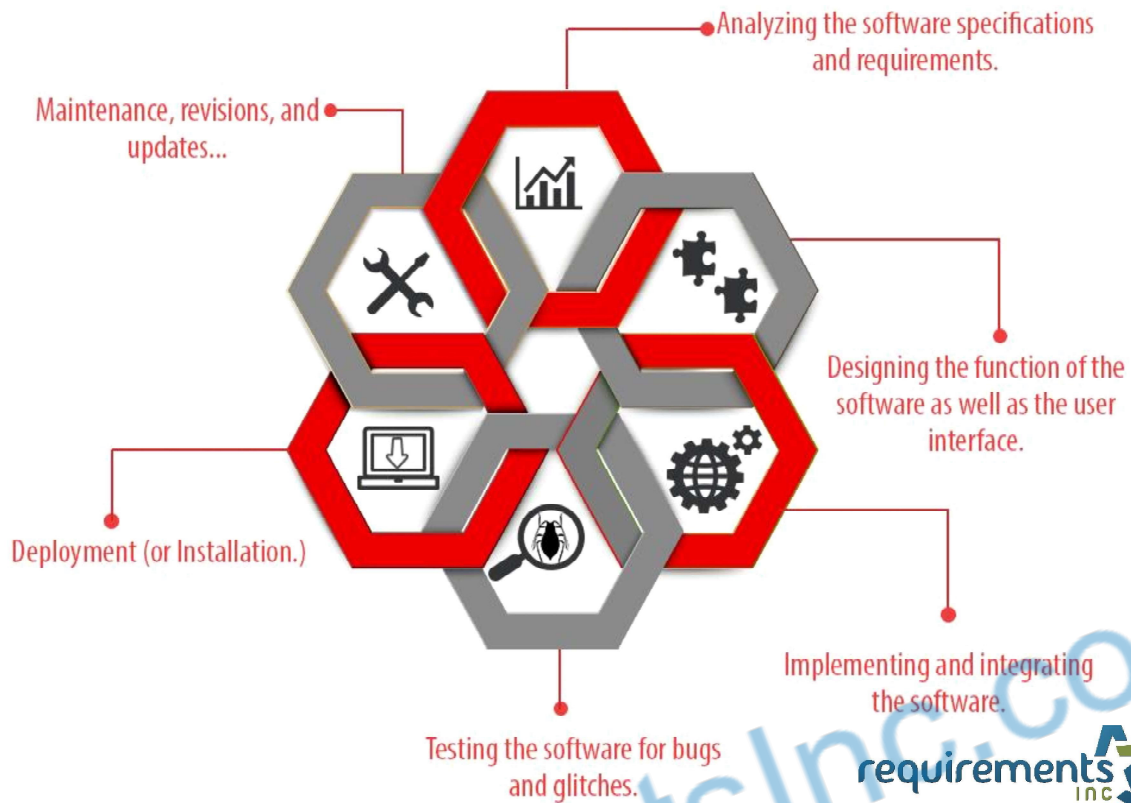
Maintenance

(Primarily, this involves receiving user feedback and making improvements to the design or the code of the software.

It could also include consultation, re-installation, and possibly some training.)



Key principles of the WATERFALL METHODOLOGY are...

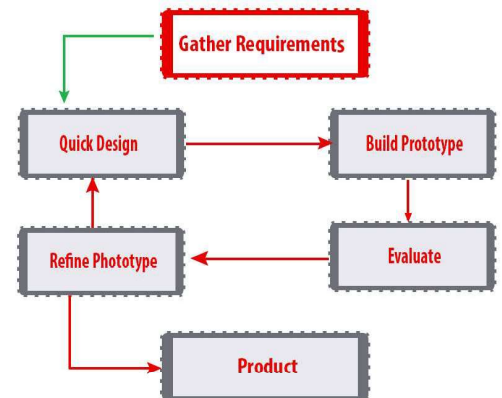


The series of software releases are known as **"increments."** Each increment is intended to provide more functionality to the users of the program

This model lets software developers learn from development of earlier versions to improve new versions.

After the first increment, a customer can use the software and provide initial feedback for the 2nd iteration.

The process continues until the product is absolutely useful and functional.



The basic idea behind this method...



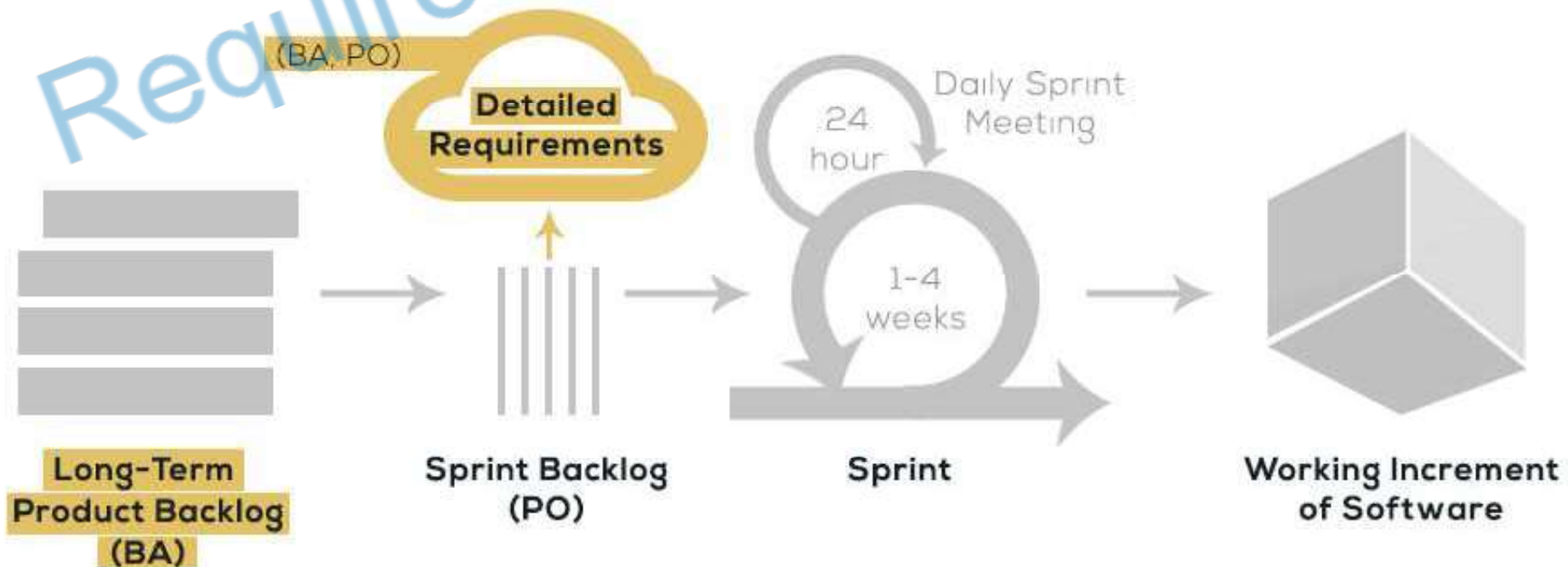
To develop a system through repeated cycles... (iterative)

and in smaller portions at a time... (incremental)

ITERATIVE & INCREMENTAL MODEL

PROPOSED AGILE PROCESS FLOW

HOW BA TEAMS CAN WORK EFFECTIVELY WITH AGILE DEVELOPMENT



BA = BUSINESS ANALYST, PO = PRODUCT OWNER

AGILE TERMINOLOGY



USER STORY

Short (1-2 sentences) description of a feature that needs to be developed, written from the perspective of a user.

**USER STORIES ARE
POINTERS TO DETAILED
REQUIREMENTS.**

SPRINT

A short timeframe (usually lasting 1-4 weeks) during which a useful increment of software is developed.



SPRINT BACKLOG

Collection of user stories being developed during a given sprint.



SCRUM

The most popular Agile methodology used by Development teams.



PRODUCT OWNER (PO)

Person who works closely with Development team during sprint, and acts as final authority representing customer interest.



PRODUCT BACKLOG

All user stories for a given product, including stories that are not yet assigned to a sprint.



**BAs FOCUS ON
LONG-TERM PRODUCT
BACKLOG, WHEREAS POs
FOCUS ON SHORT-TERM
SPRINT BACKLOG.**

SCRUM MASTER (SM)

Person who makes sure the Development team follows Agile/Scrum processes during the sprint.

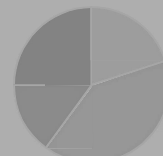
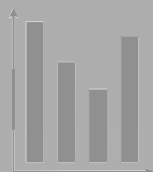




2 OBJECT-ORIENTED ANALYSIS AND DESIGN (OOAD)

2.1 Object Orientation Basics

LEARN HOW OO MINDSET CAN HELP EFFICIENT ANALYSIS
UNDERSTAND THE FUNDAMENTAL BUILDING BLOCKS OF OOAD



Benefits of Object-Oriented Analysis

- A tool to recognize, define and discuss capabilities pertaining to 'important things' involved in a project
- Helpful in everyday analysis and allows better requirements structure (avoid repetition, maintain documents adaptable to change)
- Helps collaborate better with programmers to derive an elegant solution to a business problem

FYI OO Helps you with the a Classic BA Problem: How to 'get into the heads' of your stakeholders, extract what they know about their world (a business system) and pass it along to the developers, who need to replicate that on an automated system (a software system)?



Business Analyst Workshop



Endorsed Education Provider
Endorsed Program™

15

Why Objects form the Basis of Analysis

- An Object is the basic unit by which we organize our thoughts and knowledge - "things"
 - Identify your key objects: "key objects" (a health insurance policy)
 - Define their characteristics: "data" (policy premium amount, expiration date)
 - Understand their capabilities: "functions" (pay for medical visits, pay for emergency room visits, pay for prescriptions)
 - Realize the relationship between the objects: "how things relate" (how does a health insurance policy relate to a customer? Can a customer have multiple policies?)

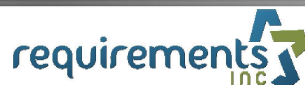
KEEP IN MIND

Objects may become apparent from discussions with end users

- they come up repeatedly and frequently
- users may feel the need to track their status



Business Analyst Workshop



Endorsed Education Provider
Endorsed Program™

16

Objects

Cookie

Laptop

Chair

Customer

Credit Card

Email

Receipt

Alert

Itinerary

Loan

Bank Account

Donation

Employment

Reservation

TANGIBLE
OBJECTS

CONCEPTUAL
OBJECTS

What constitutes an object?

Object – has its own characteristics (data) & capabilities (functions)

Data – Characteristics/properties of any object (aka attributes, fields, variables, values or states)

Functions – An object has operations, that can be performed, built right into it (aka operations, behavior or methods)



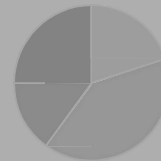
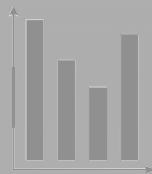
Examples of Objects

Object	Data	Functions
Cookie Cutter	Size; Material; Shape	Cut()
Laptop	RAM Capacity; Hard Disk Capacity; Screen Size	Compute(); Display(); Process()
Chair	Dimensions, Color, Price	Sit()
Customer	Name, Address, DOB	Buy(); Call(); Set up alert(); Complain()
Credit Card	Card #, Type, Expiration Date	Authorize charge()
Email	Size, Sender, Receiver, Sent Date/Time	Communicate()
Receipt	#, Date, Subtotal, Tax, Total	Confirm purchase()
Alert	Date, Reason, Priority	Monitor critical event(); Generate()
Itinerary	Departure Date/Time, Flight #	Print(); Update()
Paystub	Pay Date, Pay Period, Gross, Net	Approve(); Generate Loan Note(); Remit(); Send monthly Invoice()
Bank Account	Account #, Balance, Type	Credit(); Debit()
Donation	Amount, Beneficiary Name	Generate tax forms(); Send thank you email()
Employment	Start Date, Designation	Create offer letter(); Generate benefits statement()
Reservation	Confirmation #, Date	Confirm reservation with airline(); Confirm reservation with car rental()



BREAK OUT Exercise

SPLIT INTO TEAMS, TAKE 10 MINS



Break Out Exercise



IDENTIFY OBJECTS FOR JOE

Joe joins a new project to build a loan origination system. In one of his initial meetings with business users, he takes notes to document key aspects of the business that may be critical to the project (noted on the right side). Can you identify the "objects" on the list?

- Loan
- Borrower
- Co-borrower
- Document Custodian
- Borrower Account
- Customer Service Agent
- Principal
- Interest Rate
- Interest calculation logic
- Different types of loans
- Equifax, company used for credit check
- Loan terms
- Payment
- Loan Note
- Borrower account
- Customer service agent
- Late payment fees

TIP Attempt to identify at least one data and one method for an item. If you can, the item should be an object!



Let's get a bit specific!

- An Object represents a **single, uniquely identifiable** thing
- Example** A car object: Make=Toyota, Model=Camry, Year=2007, Color=Black, Transmission type=Manual, Drive type=Front Wheel, # of doors=4, VIN#=JH4NA1152MT001365 etc.
- Objects can "access" each other's functions by sending/receiving **messages**.

Messages can also "pass" desired 'data'

- iPodObject.Play(7th song on playlist)
- Transaction.Withdraw(500) accesses Account.Debit(500) resulting in a change to Account.Balance from 5000 to 4500
- Transaction.CheckBalance(Checking) accesses Account.GetBalance() to display Account.Balance as 4500

Some methods may not pass data

Object – A unique thing that has its own characteristics (data) and capabilities (functions). Objects interact with each other via messages, that allow them to access other object's functions.



Business Analyst Workshop

requirements
inc

IBA®

Endorsed Education Provider
Endorsed Program™

21

What is a Class?

- A classification of similar objects that share the same attributes, operations and relationships.

Example Automobile Class, Car Class, SUV Class, Toyota Camry Car Class

- Has same data and functions, but no specific values
 - Has data that defines it: Make, Model, Year, Color, Transmission type, Drive type, # of doors, etc.
 - Has functions that it can perform: drive(), race(), transport(), tow(), etc.
- Objects are created from a Class. Class is the template (car specifications/design), Object is the product (car)

Automobile



TIP

To remember, think of Class as a "Cluster" or a "Collection" or a "Classification"(of objects)



Business Analyst Workshop

requirements
inc

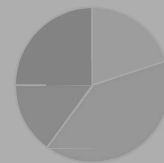
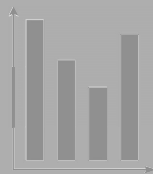
IBA®

Endorsed Education Provider
Endorsed Program™

22

2.2 OO Concepts

LEARN INSIGHTS THAT A BA CAN DERIVE OUT OF KEY
OBJECT ORIENTED CONCEPTS



Encapsulation or Data Hiding

“Less is more”

Learning to drive is easy enough to learn/teach since you do not need to understand the internal workings - plus, it safeguards vehicles from user errors that can wreck the internal mechanics. Since you only need to know about limited interfaces made visible to you (steering wheel, gear, etc.), it is seamless to move from one car to another (no need to relearn driving).



LESS IS MORE



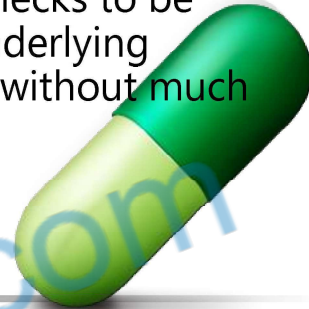
Encapsulation or Data Hiding

BA Takeaway

- Keep what is exposed to the user as minimalistic and simple as possible, hiding any internal logic (unless it is helpful / necessary for the user)
- Have a clear line of separation between objects

Example

Bank of America made a change to their ATM to enable checks to be scanned and printed on the receipts. There was a lot of underlying changes, but the process was seamless for users to adopt without much training.



Business Analyst Workshop

requirements
inc

IIBA

Endorsed Education Provider
Endorsed Program™

25

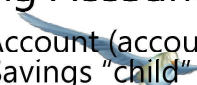
Generalization (Inheritance)

"Birds of the same feather flock together"

Similar classes can be grouped together. Other variations can be created by inheriting the behavior of the original class. Helps reuse rather than reinvent.

Example

- Pen (general) ← Fountain Pen (specific)
 - Fountain pen has all characteristics/functions of a Pen
- Account (general) ← Checking Account, Savings Account
 - All attributes from the "parent" Account (account #, date opened, balance, etc.) are available to both Checking and Savings "child" accounts.



Business Analyst Workshop

requirements
inc

IIBA

Endorsed Education Provider
Endorsed Program™

26

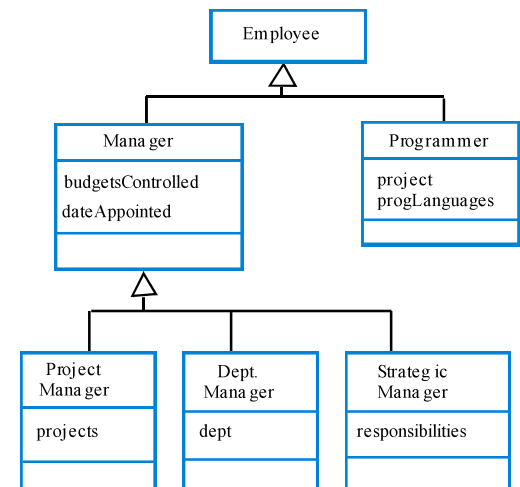
Generalization (Inheritance)

BA Takeaway

- Helps document at the 'right level', avoid redundancy

Example

If a new policy allows all Managers to get some additional perks, you can add it to the Manager class, rather than each type of manager. This will affect any new types of managers that get added later seamlessly.



Parent (base/superclass)

- Provides common functionality and data members

Child (derived/subclass)

- Inherits public and protected members from the superclass
- Can extend behavior

TIP

Use Child "is a kind of" Parent test to identify inheritance relationship. Manager "is a kind of" Employee. Project Manager "is a kind of" Manager



Abstraction

“Know your audience”

Level of detail presented based on the intended audience, the context, the stage of the project, etc.



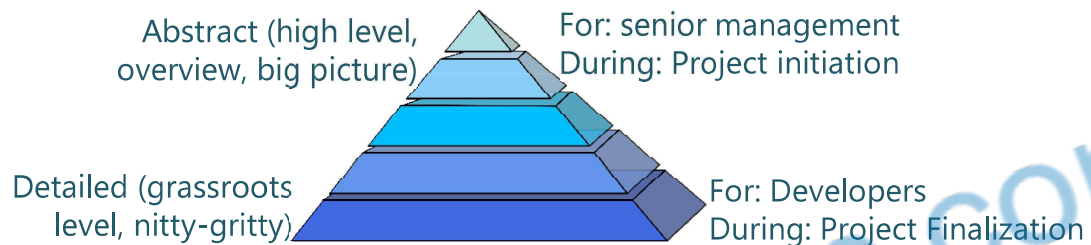
Abstraction

BA Takeaway

- Allows moving from a high level overview to the lower level detail
- Allows us to first present a high level to an end user to set context before jumping into the detail

Example

- You can describe how to use a coffee machine in 3 steps (to someone who is familiar) or in 20 steps (to someone who is new)
- Online Banking / Bank Statement: Monthly summary view is an abstraction of detailed transactions view



Business Analyst Workshop

requirements inc

IBA

Endorsed Education Provider
Endorsed Program™

29

Association Relationship

"No man is an island"

Objects are meaningful/useful in relationship with other objects

BA Takeaway

- It is critical to understand how the different objects relate to each other

Example

- Which other objects are interacting with an "invoice" object? Payer, payee, bank that processes the payment
- Statement object is associated to Customer Class, Account Class and Transactions Class, to generate bank statements.



Business Analyst Workshop

requirements inc

IBA

Endorsed Education Provider
Endorsed Program™

30

Aggregation Relationship

“Like a school of fish”

Defines the relationship of objects that make up other objects. Remove some smaller objects and the larger object is still meaningful.

BA Takeaway

- It is useful to identify how related business objects are structured.

Example Which departments make up a company?



Business Analyst Workshop

requirements
inc

IBA

Endorsed Education Provider
Endorsed Program™

31

Composition Relationship

“Like the pieces of a puzzle”

Defines the relationship of objects that make up other objects. Remove a smaller object and the larger object disintegrates.

BA Takeaway

- Useful to establish all the key pieces that make up an object.

Example An order consist of customer info, billing info, shipping info, product list, quantity for each product, confirmed payment (order cannot be considered complete without all this info)



Business Analyst Workshop

requirements
inc

IBA

Endorsed Education Provider
Endorsed Program™

32

Multiplicity Relationship

“Numbers matter”

Defines the rules of cardinality

BA Takeaway:

- Certain rules may need to be enforced in a system. These requirements can be elaborated using the multiplicity concept

Example

- One rebate per household
- A promotion requires the user to purchase 3 or more items.
- Each subject must be handled by at least one teacher



Business Analyst Workshop



Endorsed Education Provider
Endorsed Program™

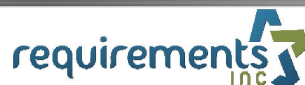
33

What is UML?

- Widely used Object-Oriented visual modeling language for system design. Helps get everyone on the same page.
- BAs use UML to
 - Sketch out ideas
 - Describe business processes
 - Capture system behavior
 - Depict relationship between objects
- Developers / Database Analysts / Solution Architects use UML to
 - Show underlying application structure / system architecture
 - Depict communication between objects
 - Model data structure
- UML has
 - Notations – elements that work together in a diagram: symbols & connectors
 - Diagrams – pictorial representations of the system, process or some parts of the system



Business Analyst Workshop



Endorsed Education Provider
Endorsed Program™

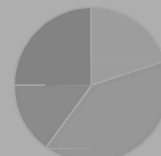
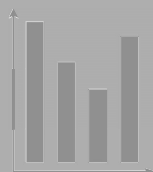
34





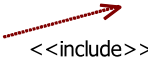
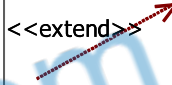
3 UML USE CASE DIAGRAMS

3.1 Use Case Basics & Notation

RECOGNIZE HOW USE CASES DESCRIBE
FUNCTIONAL COMPONENTS OF A SYSTEM

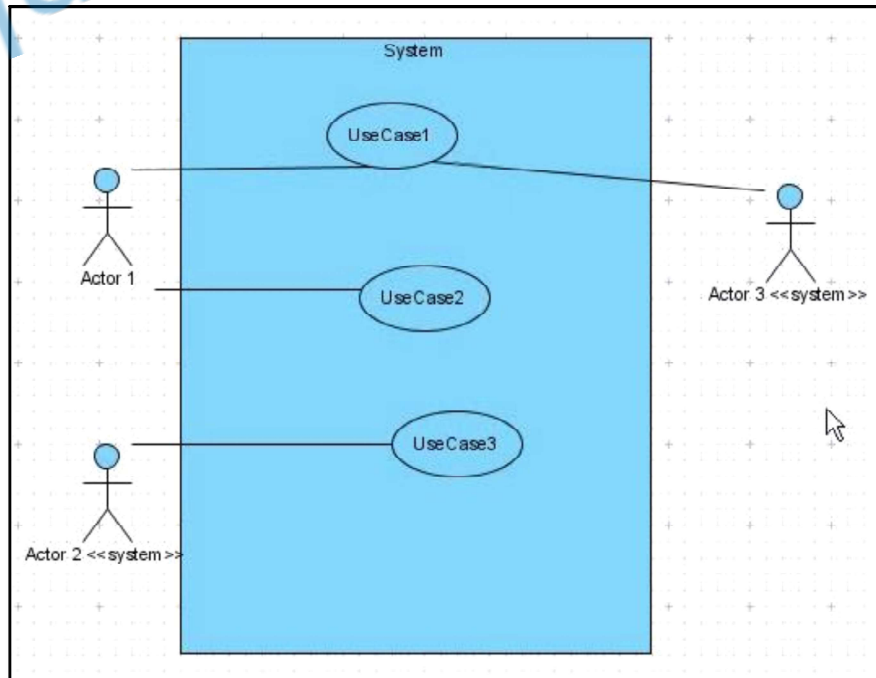


Use Case Model Notation

Association	Associations are used to show interactions. (Undirected) associations are depicted by a plain line. Primary actors on the left and secondary actors on the right if using plain association (Semantically same as directed association)	
Generalization	Inheritance relationship (generalization in UML) is indicated by an arrow with a filled arrow head pointing to the parent.	
<<Include>> Stereotype	A <u>base use case</u> executes an <<include>> use case as part of its flow to complete its business functionality. E.g., <u>Transfer Balance</u> <<includes>> Check Balance	
<<Extend>> Stereotype	A <u>base use case</u> may be extended by an <<extend>> use case based on a certain condition. E.g., Buy extended warranty <<extends>> <u>Buy laptop</u>	



Sample Use Case Structure

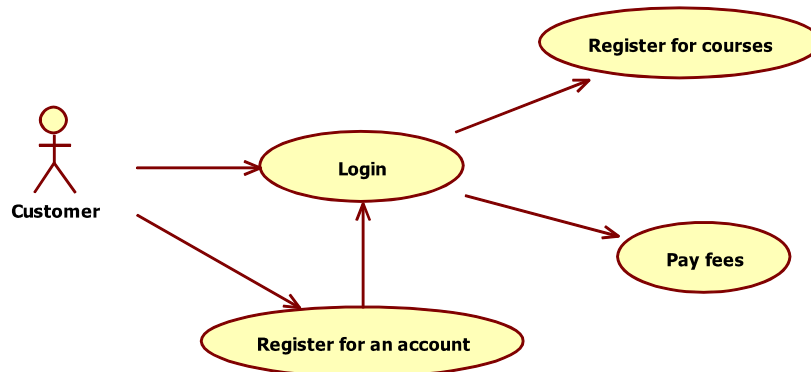


Preconditions

- Preconditions are statements that must be true (or other use cases that must have executed) before the use case in hand is executed

Example

- “Register for an account” is a precondition to “Login”
- “Login” is a precondition to “Pay fees”



Business Analyst Workshop

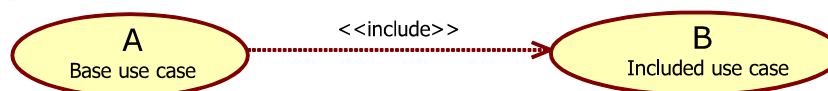
requirements inc

IBA

Endorsed Education Provider
Endorsed Program™

51

<<Include>> Construct



- Signifies that the base use case calls the included use case (also <<uses>> or <<includes>>)
 - A includes B
 - A is the base (calling) use case, B is the included (called) use case
 - During A's execution, B will be called one or more times and then control comes back to A
 - A cannot produce success outcome without running B
 - Nothing precludes B from being executed directly by the user/another use case without <<include>> relationship
- Example: Transfer Funds <<includes>> Check Balance; Check Balance may be executed by itself as well



Business Analyst Workshop

requirements inc

IBA

Endorsed Education Provider
Endorsed Program™

52

Break Out Exercise



SIMPLE FLIGHT RESERVATION

Here are a few actors

- Customer
- Payment Processor

Here are a few use cases

- Search flights
- Make a reservation
- Purchase Ticket
- Check Flight Status
- Cancel Flight
- Reschedule Flight (option presented when you cancel flight only)
- Validate Credit Card
- Select seat (option available when purchasing tickets only)

Come up with a use case diagram (use pen & paper)



Business Analyst Workshop

requirements
inc

IBA®

Endorsed Education Provider
Endorsed Program™

67

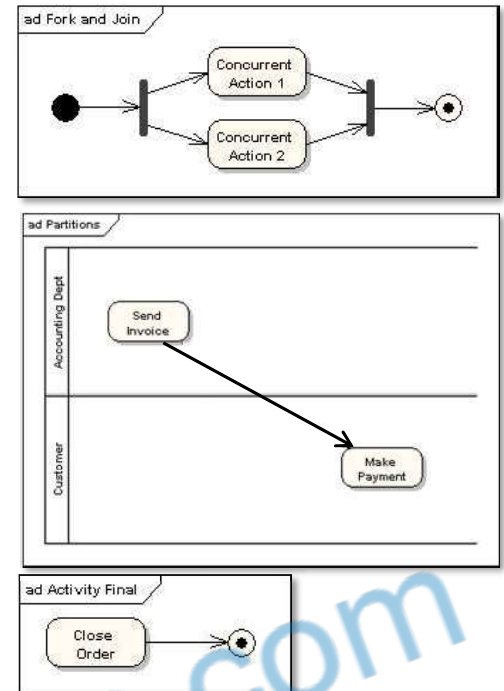
requirements
inc



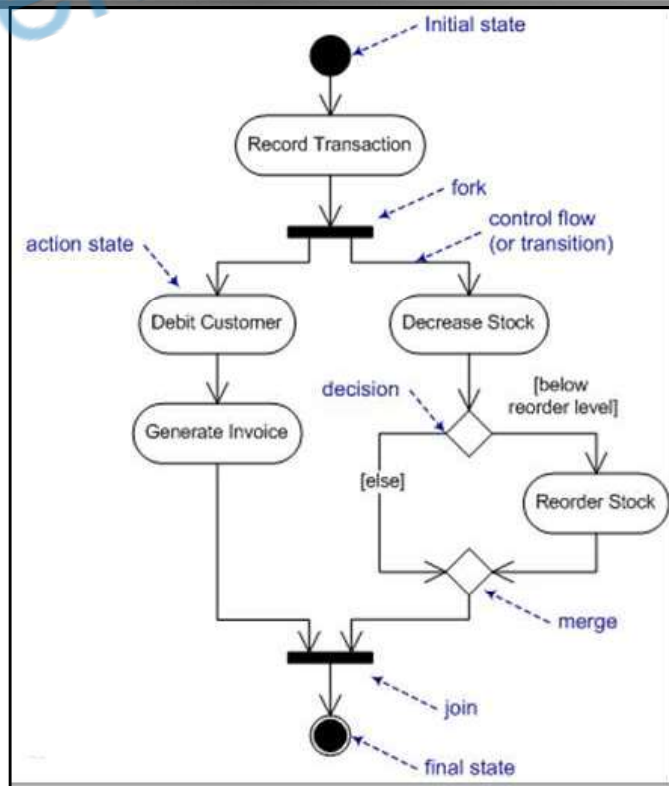
4 UML ACTIVITY DIAGRAMS

Activity Model Notation

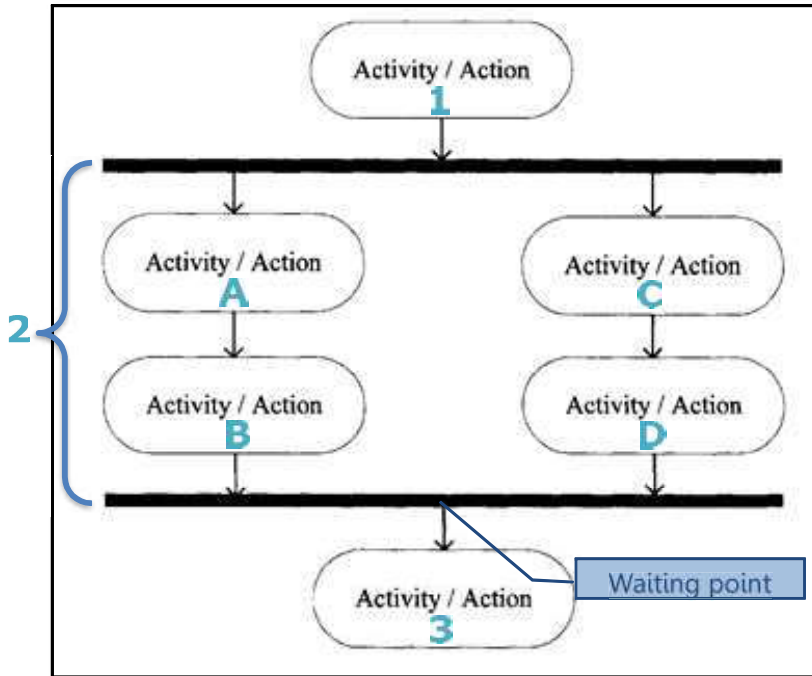
- **Fork/Join:** Drawn as parallel lines, the former line being fork, and the latter, join. Steps depicted 'side by side' between the fork and the join may follow any sequence.
- **Swimlanes:** Depicts the actor that performs the step
- **Final Node:** End point of an activity. An activity diagram can have multiple final nodes (since each path may end with a different outcome).



Activity Model Notation



Fork / Join Example



- 2A must be completed before 2B
- 2C must be completed before 2D
- 2A-2B and 2C-2D can be completed in any order. For instance, 2D may be completed before 2A.
- 3 can be completed only after 2A, 2B, 2C and 2D are completed



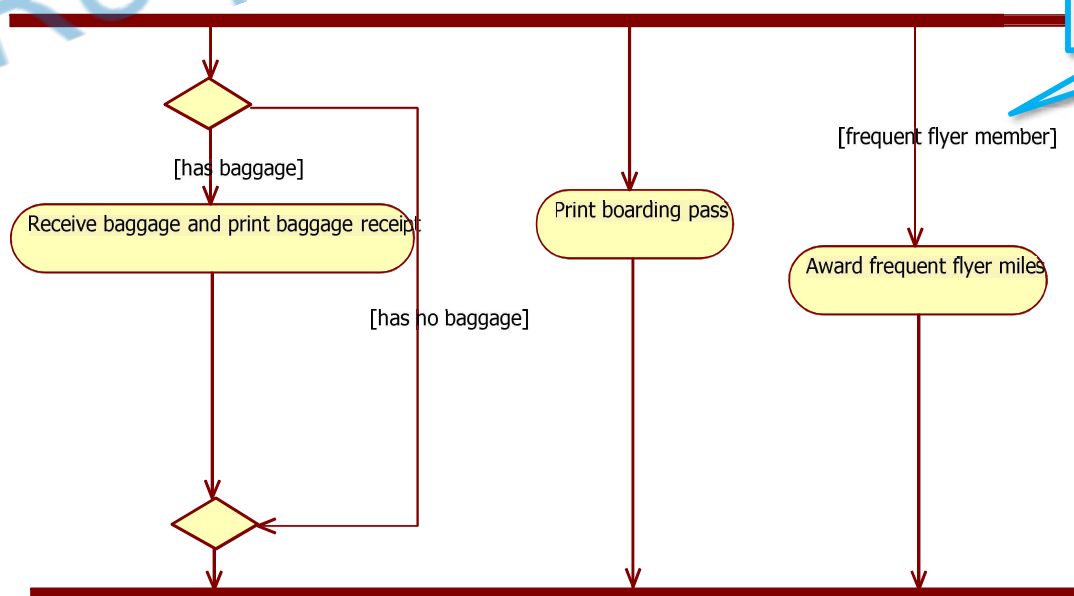
Business Analyst Workshop



Endorsed Education Provider
Endorsed Program™

81

Fork / Join Example



Shortcut method -
Same as using a
decision point



Business Analyst Workshop

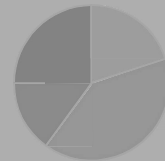
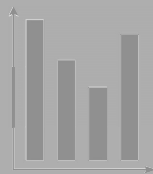


Endorsed Education Provider
Endorsed Program™

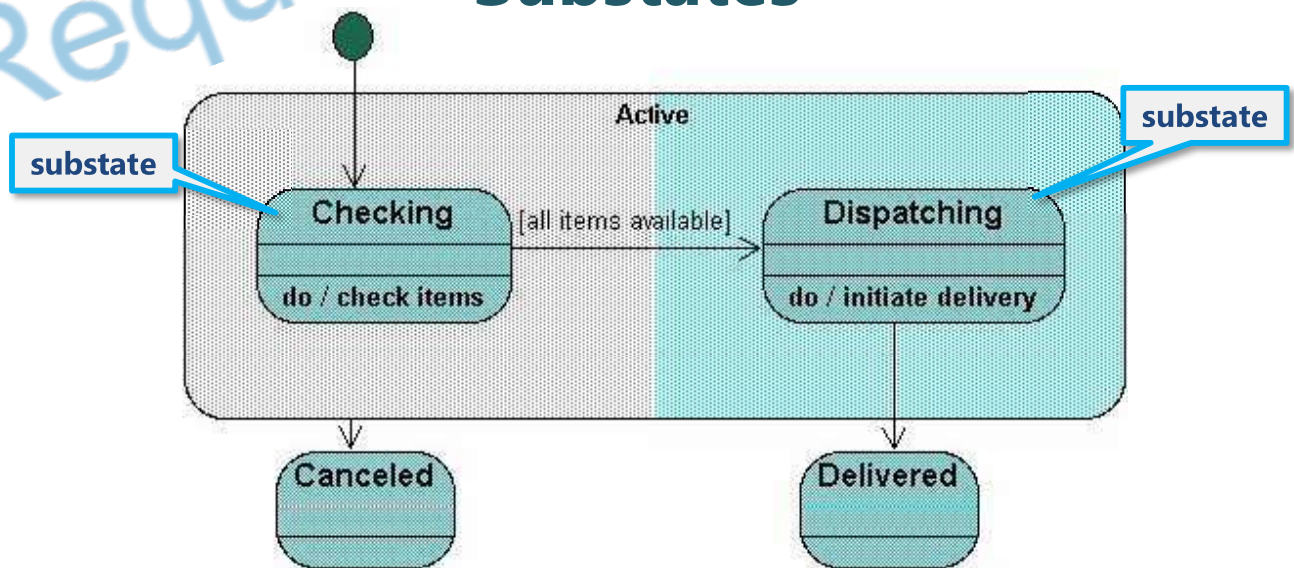
82

5.2 Substates

EVALUATE HOW SOME STATES HAVE MORE SPECIFICS,
DENOTING SUBSTATES

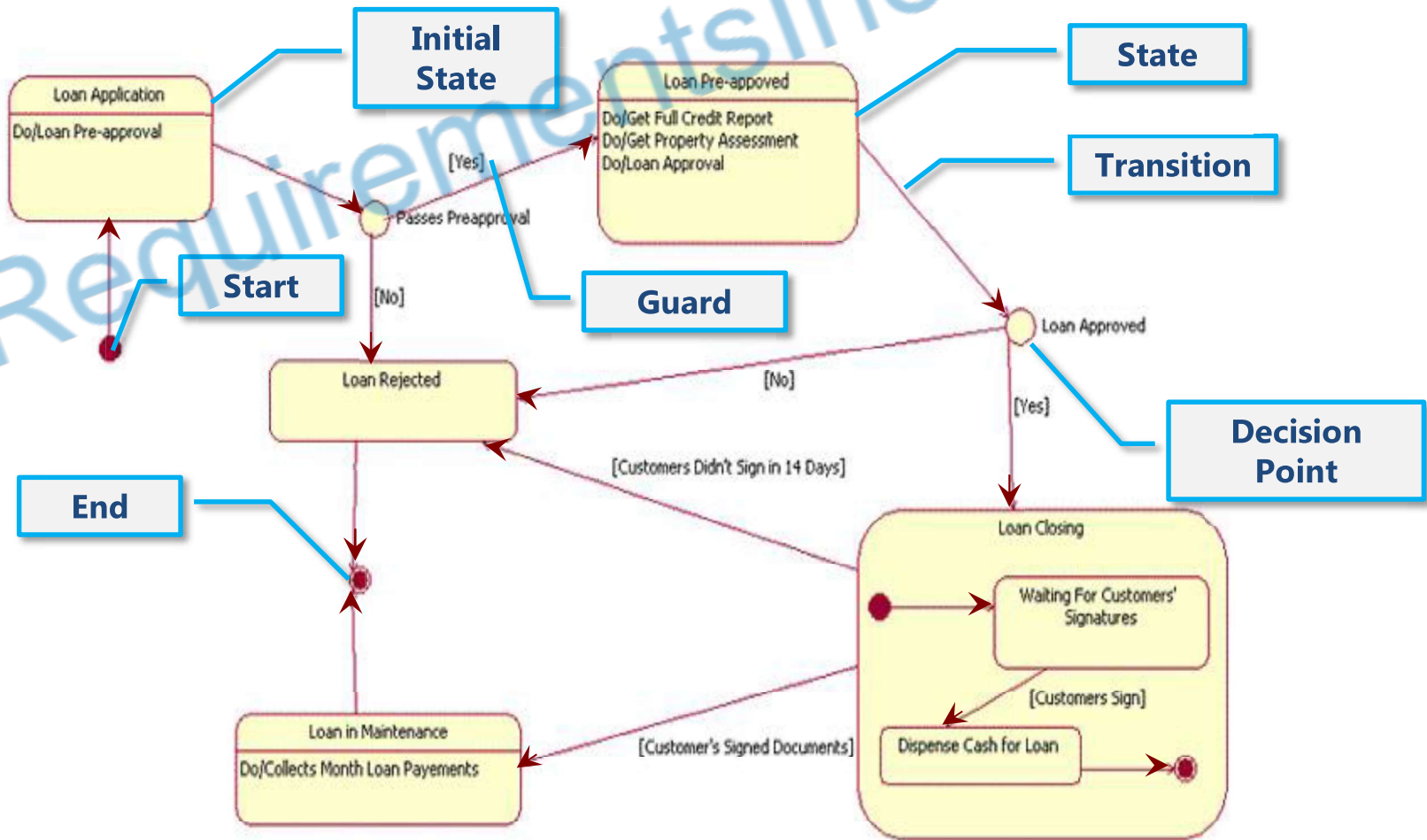


Substates



TIP

Substate gives you more specific info about an object within a state. We use substate when the state needs to be revealed to one person (say customer) and more detailed substate needs to be revealed to another (internal employee).



Break Out Exercise



IDENTIFY STATES

- Think about the states for an ATM.
- Keep in mind the entire lifecycle
 - While the customer is interacting
 - While the ATM is idle



Business Analyst Workshop

requirements
inc

IBA®

Endorsed Education Provider
Endorsed Program™

123

requirements
inc



6 REQUIREMENTS

Requirement Types

- **Business Requirements** (solution to the business problem)
 - Address the business problem or goal.
 - Identify why the project is being undertaken.
 - Lead an organization to increase revenue, decrease costs, or enhance service.
- **User Requirements**
 - Address user-related issues, needs, or expectations.
 - Identify how the user will interact with the system.
 - Express from the user's point of view.
 - User Requirements are driven by the tasks an end-user needs to perform or a capability they want.



Requirement Types

Functional Requirement defines a function of a system and its components. A function is described as a set of inputs, the behavior, and outputs.

- "what can the user achieve"

Example

Send an email when a condition is met (e.g. an order is placed, a customer signs up)

Non-Functional Requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

- "how do we define the quality of the system?"
- Performance (example, Response Time), Scalability, Capacity, Availability, Reliability, Security, Usability, Interoperability

Example

Emails should be sent with a latency of no greater than 12 hours from the generating activity



Think SMART

- **Specific** – unique, necessary, precise, concise, unambiguous
 - Avoid conjunctions (and, or, but) or absolutes (“all”, “never”, “always”)
 - Avoid indeterminate amounts of time (soon, fast, later, immediately)
- **Measurable** – testable, complete
 - Avoid non-fact based measurements such as “best” or “optimal”.
- **Attainable** – feasible, appropriate
- **Realistic** – relevant, consistent
- **Traceable** – identifiable, linked to root cause / need
 - Why does a requirement exist? To satisfy who? What happens when you remove the requirement?

Why good requirements matter?

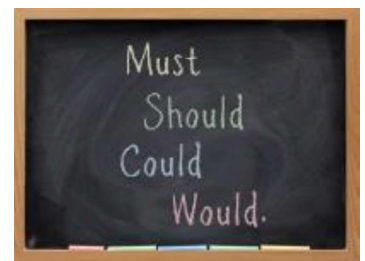
- Improve product quality (Satisfies the end users)
- Reduce rework (Makes for a happy project team)
- Deliver on time and budget projects (Satisfies the sponsors)



Prioritizing Requirements

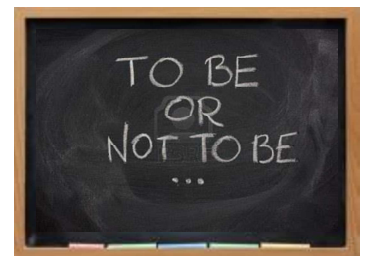
MoSCoW stands for must, should, could and would:

- **M** - Must have this requirement to meet the business needs.
- **S** - Should have this requirement if possible, but project success does not rely on it.
- **C** - Could have this requirement if it does not affect anything else in the project.
- **W** - Would like to have this requirement later, but it won't be delivered this time.



Must Have / Nice to Have

- **Must-Have** - Absolutely has to be delivered for the project to be considered successful.
- **Nice-to-Have** - Desired or even important to the overall deliverable, but can be considered as optional or nice-to-have in the overall completion of the project.



JAD Activities

Working Session

- Set-up stage –
 - Welcome participants
 - Presents tasks and schedule
 - Establish rules
 - Define what is off topic
- Brainstorming
 - Achieve consensus on decisions
 - Generate ownership of results
 - Create any deliverables
- Identify open issues and questions

Summary

- Follow-up
 - Resolve open issues and questions
 - Follow-up on action items
- Wrap-up
 - Review results of follow-up items
 - Evaluate the JAD process
 - Discuss "lessons learned"
 - Finalize deliverables



7 WRITTEN USE CASES

Supplemental Specifications

- Negotiable guidelines that may not affect business policies and procedures
 - Number of records to be displayed on a page
 - Text content in an email notification
 - Sort Order
- SS Template
 - Spec Number (unique id)
 - SS39
 - Spec Description
 - 25 rebates must be presented in a single view
- Documented separately out of other requirement (use case flow) and linked to the use cases (reused, can form a collection or 'specs' of the business)



Main scenario (happy path)

1. System presents the customer with the list of available phone cards
2. User selects to purchase a phone card
3. The system provides the following options:
 - Purchase by logging in
 - Purchase without login (if customer chooses this option, Alt 1) **BRANCH OUT OF MAIN FLOW→**
4. User logs into the system using the username and password (if new customer Alt 2) **BRANCH OUT OF MAIN FLOW→**
5. System validates the username, password successful (if login fails Exc 1) **BRANCH OUT OF MAIN FLOW→**
6. User places an order by adding phone card(s) to shopping cart
7. User provides credit card information and billing address
8. System contacts payment gateway to authorize charge (if authorization fails Exc 2) **BRANCH OUT OF MAIN FLOW→**
9. System confirms sale
10. System presents the phone card PIN number(s) and sends a confirmation email with the PIN number(s)

Alternate scenario 1

1. System requests the user to enter full name, email and phone number.
- Use case returns to step 6 of the main flow. **←MERGE BACK TO MAIN FLOW**

Alternate scenario 2

1. System prompts the user to enter name, email, default mailing address, password, confirm password
 2. User enters the required information
 3. System records the email address as the username and confirms account.
- Use case returns to step 6 of the main flow. **←MERGE BACK TO MAIN FLOW**

Exception scenario 1

1. User is presented with a login error message
- Use case ends.

Exception scenario 2

1. User is presented with an error message that the credit card has failed and the error reason
- Use case ends.



Use Case Guidelines

If a use case is relatively too long, consider

- Horizontal split
 - Find a place to logically stop one business process and start another one
- Vertical split
 - If several alternates for a use case are similar, those alternates can be combined into another use case

E.g., If a 'login' use case has several flows that are related to forgotten login info, i.e.,

- forgotten password retrieved via email, answering security q and a or by verifying personal information,
- forgotten username retrieved via email or by verifying personal information, then the login use case may be split into 'login' and 'retrieve login'



Business Analyst Workshop

requirements
inc

IBA®

Endorsed Education Provider
Endorsed Program™

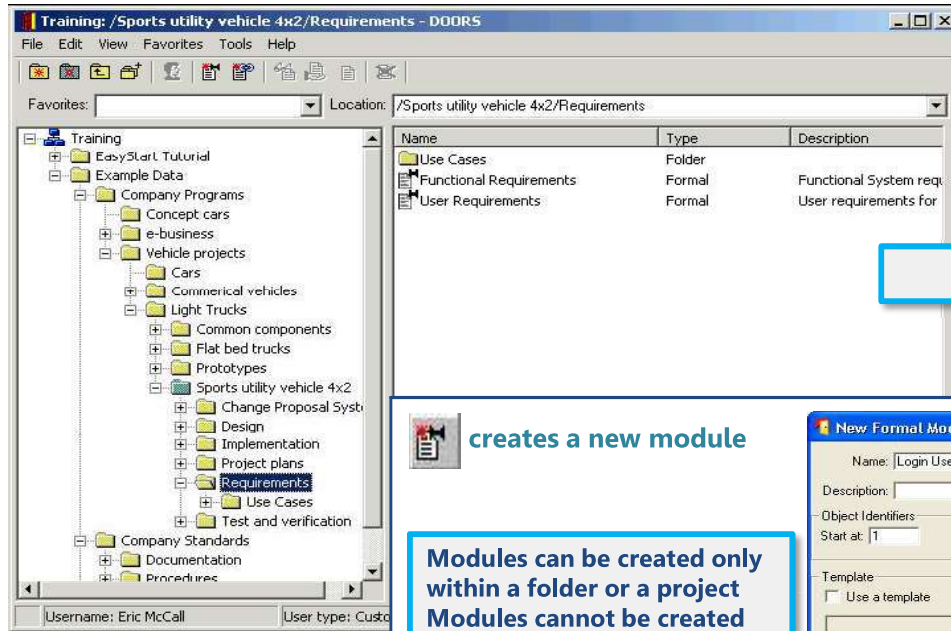
155

requirements
inc



8 IBM DOORS FOR REQUIREMENT MANAGEMENT

Projects, Modules and Folders



Project
Folder
Module

DOORS
Explorer view

creates a new module

Modules can be created only
within a folder or a project
Modules cannot be created
under the root directory



Business Analyst Workshop

requirements
INC

IBA

Endorsed Education Provider
Endorsed Program™

161

Objects and Attributes

Columns = Attributes

Rows = Objects

User requirements for SUV 4x2	Priority	Approved
3.1.4 Fuel economy	Medium	No
Users shall be able to obtain fuel consumption better than that provided by the 95% of cars built in 1996.		
3.1.5 Safety	Low	No
Users shall be able to accelerate from 0 to 100 kilometers per hour in 10 seconds.		
3.1.5 Safety	Low	No
Users shall be able to accelerate from 0 to 100 kilometers per hour in 8 seconds.		
3.1.5 Safety	High	Yes
Users shall be able to travel in safety in accordance with the Road Research Laboratories Safety standards dated 1 January 1993.		
3.1.5 Safety	Yes	Yes
Users shall be able to travel at the same level of safety		



- Requirements are documented as objects
- Additional information about requirements are documented as attributes
- Attributes are useful to sort, filter, search, save existing view
- Default attributes like 'Created By' and 'Modified On' are available
- Can create other attributes to "tag" or "categorize" requirements



Business Analyst Workshop

requirements
INC

IBA

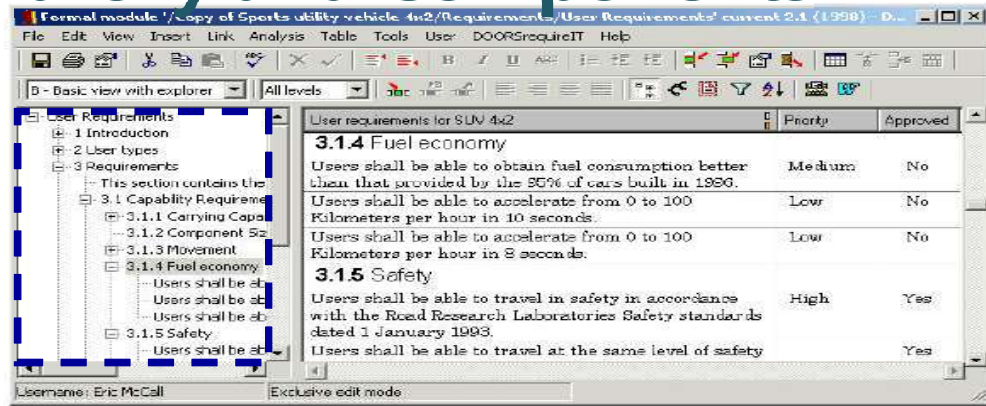
Endorsed Education Provider
Endorsed Program™

162

Objects Hierarchy and Components

- Object hierarchy facilitates organizing the headings and grouping similar objects together
- Similar to headings and sections in Word (View-> Document Map)

Objects Hierarchy



An object can have an object heading & object text

3.1.5 Safety

Users shall be able to travel in safety in accordance with the Road Research Laboratories Safety standards dated 1 January 1993.
Users shall be able to travel at the same level of safety

High

Yes

Yes

Heading

Text



Business Analyst Workshop

requirements
INC

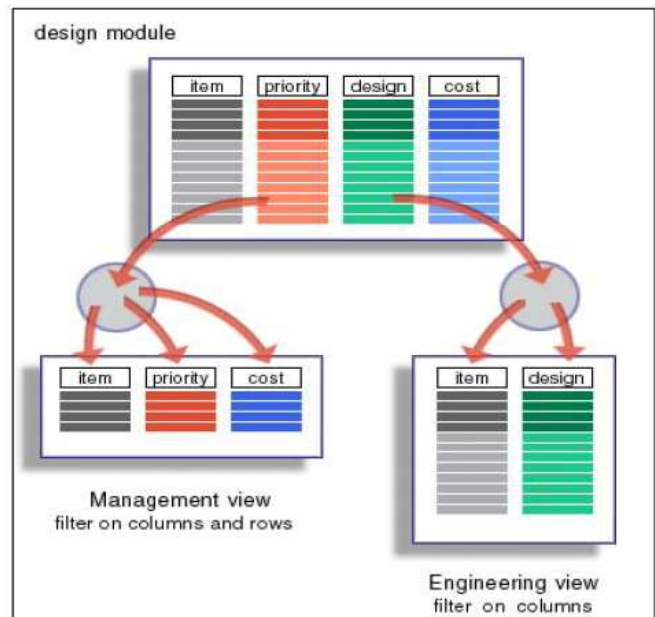
IBA

Endorsed Education Provider
Endorsed Program™

163

Views

- Views are filters that enable different information for different people
- Each view contains a subset of the objects or attributes in the module
- Views can filter out objects or attributes or both.



Business Analyst Workshop

requirements
INC

IBA

Endorsed Education Provider
Endorsed Program™

164

Creating Links

ID			
3a.	User	Chooses to register an account.	
3b.	System	Prompts to enter account information.	
		UserAccount.AccountNumber	
		UserAccount.T	

Link created
(inlink at destination)

ID			
BR1	1 Mandatory Fields	The following fields are mandatory:	
		<ul style="list-style-type: none"> UserAccount.AccountNumber UserAccount.TaxID UserAccount.DateOfBirth if UserAccount.TaxID provided is SSN UserAccount.Address UserAccount.Email UserAccount.PrimaryPhone 	
		<ul style="list-style-type: none"> UserAccount.DateOfBirth UserAccount.Address UserAccount.Email UserAccount.PrimaryPhone 	

Link created
(outlink at source)

Can navigate between
inlink and outlink

ID			
BR1	1 Mandato	/Mortgage Project/Requirements Test/Login Use Case	354: System
		The following fields are mandatory:	
		<ul style="list-style-type: none"> UserAccount.AccountNumber UserAccount.TaxID UserAccount.DateOfBirth if UserAccount.TaxID provided is 	



Business Analyst Workshop

requirements
INC



Endorsed Education Provider
Endorsed Program™

177

requirements
INC



9 JIRA FOR CHANGE MANAGEMENT

Requirements Inc. com

JIRA Dashboards Projects Issues More Create Issue Quick Search

IRKD / IRKD-46 Coworker's terrible music is audible through his headphones

Edit Comment Assign More Open In Progress Workflow Export

Details

Type:	Bug	Status:	Open
Priority:	Minor	(View Workflow)	
Affects Version/s:	None	Resolution:	Unresolved
Component/s:	None	Fix Version/s:	None
Labels:	noise		
Rank:	286		

Description

My coworker wears expensive headphones but is playing his Death Country Metal Emo music so loudly that I can make out the lyrics, even though they're in Swedish and I don't speak Swedish.

Activity

All Comments Work Log History Activity Source Reviews

Commits Builds

▼ Jesse Katz added a comment - a few seconds ago

Dan Radigan Dan, can you either pull the plug on this guy's music player or give him some early-period Miles Davis to listen to?

People

Assignee: Unassigned Assign to me

Reporter: Jesse Katz

Votes: 0 Vote for this issue

Watchers: 1 Start watching

Add Watcher

Start Type

Dan Radigan an hour ago

Agile

View on Board

Share Notifications

Make a teammate aware of the issue. Note: Sharing will not automatically add a user as a watcher.

Link to a Comment

If you want to make sure that someone sees a particular comment on an issue, you can email them a link to that comment. Click on the permalink icon for that comment, and the link is copied. You can email anyone the link.

Notify with @mentions

Like Facebook or Twitter, if you want to pull a teammate into a conversation, use the @mention feature.


JIRA Dashboards ▾ Projects ▾ Issues ▾ More ▾ [Create Issue](#) ? ⚙️ 👤


IRKD / IRKD-46

Coworker's terrible music is audible through his headphones

[Edit](#) [Comment](#) [Assign](#) [More ▾](#) [Open](#) [In Progress](#) [Workflow ▾](#) [Export ▾](#)

Details

Type: **Epic**  Status: **Open** [\(View Workflow\)](#)

Priority: **Minor**  Resolution: **Unresolved**

Affects Version/s: **None** Fix Version/s: **None**

Component/s: **None**

Labels: **None**

Rank: **286**

People

Assignee: **Unassigned** [Assign to me](#)

Reporter: **Jesse Katz**

Votes: [Vote for this issue](#)

Watchers: [Stop watching this issue](#)

Description

My coworker wears expensive headphones but is playing his Death Country Metal Emo music so loudly that I can make out the lyrics, even though they're in Swedish and I don't speak Swedish.

Activity



[All](#) [Comments](#) [Work Log](#)

[Commits](#) [Builds](#)

▼ **Jesse Katz** added a comment

Someone needs to pull the plug on the headphones with cream cheese



Details

Type: **Epic**  Priority: **Minor** 

Affects Version/s: **None** Component/s: **None**


Labels: **None** Rank: **286**

Details

Type: **Epic**  Priority: **Minor** 

Affects Version/s: **None** Component/s: **None**

Labels: **None** Rank: **286**

Epic  Status: **Open**

- Bug**
- [New Feature](#)
- [Task](#)
- [Improvement](#)
- [Story](#)

Inline Editing
Without bringing up a dialog box or editing page, you can make changes to JIRA issues.

Requirements Inc. watermark: requirementsinc.com

JIRA Dashboards Projects Issues More Create Issue Quick Search

IRKD / IRKD-46 Coworker's terrible music is audible through his headphones

Edit Comment Assign More Open In Progress Workflow Export

Details

Type: Bug
Priority: Minor
Affects Version/s: None
Component/s: None
Labels: None
Rank: 286

Add a Label with inline edit

The type field has been changed to a bug!

In the next step, watch how to add a label with inline editing.

Labels: [input field]

Labels: noise
Rank: noise (New Label)

Labels: noise x

Description

My coworker wears expensive headphones so loudly that I can make speak Swedish.

Activity

All Comments Work Items Commits Builds

▼ Jesse Katz added a comment - 2 minutes ago

Someone needs to pull the plug on this guy's music player. Or else stuff his headphones with cream cheese.

People

Assignee: Unassigned
Assign to me

Reporter: Jesse Katz

Details

Type: Bug
Priority: Minor
Affects Version/s: None
Component/s: None
Labels: noise
Rank: 286

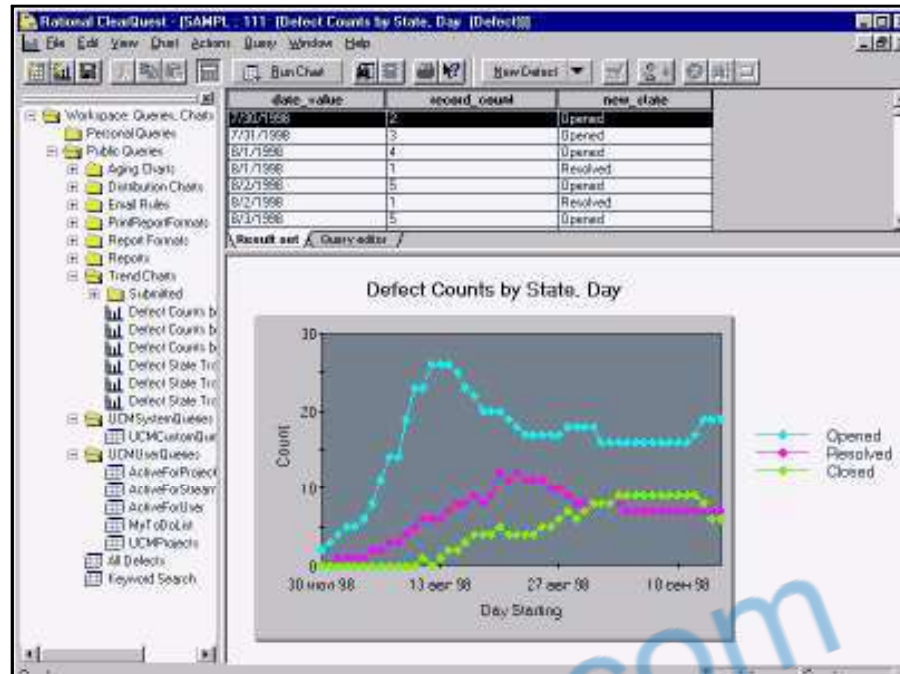
Updated: a minute ago

Agile

View on Board

Comment

- Management reports can be run based on statuses, counts, effort spent, etc.



Business Analyst Workshop

requirements
INC

IBA

Endorsed Education Provider
Endorsed Program™

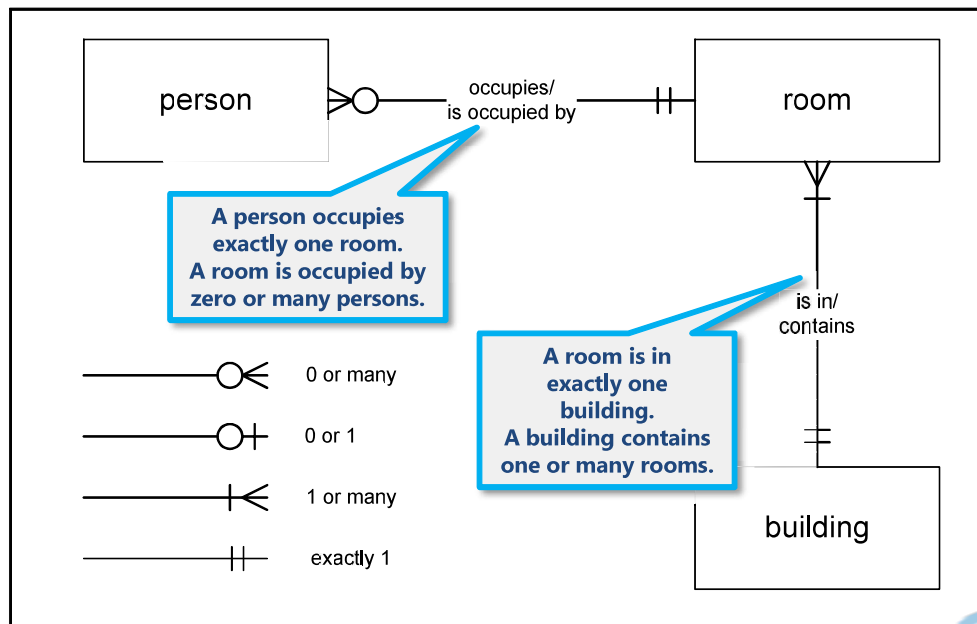
207

requirements
INC

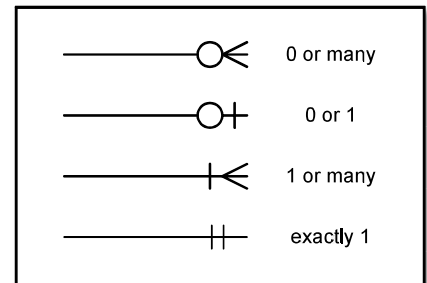
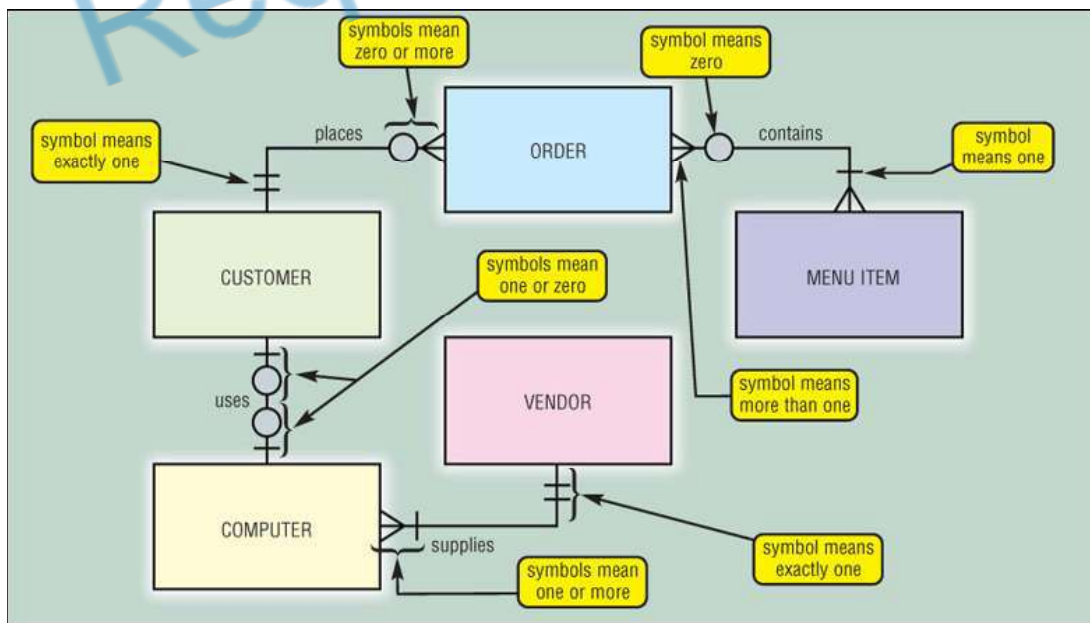


10 ENTITY RELATIONSHIP DIAGRAMS (ERD)

Example 1



Example 2

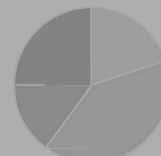
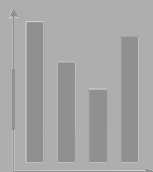




11 STRUCTURED QUERY LANGUAGE (SQL)

11.1 SQL Introduction

UNDERSTAND HOW DATA IS MAINTAINED IN THE SYSTEM



Joining data from two tables

Employees:

Employee_ID	Name
01	Hansen, Ola
02	Svendson, Tove
03	Svendson, Stephen
04	Pettersen, Kari

Orders:

Prod_ID	Product	Employee_ID
234	Printer	01
657	Table	03
865	Chair	03

Who has ordered a product, and what did they order?

```
SELECT Employees.Name, Orders.Product
FROM Employees, Orders
WHERE Employees.Employee_ID=Orders.Employee_ID
```

Result

Name	Product
Hansen, Ola	Printer
Svendson, Stephen	Table
Svendson, Stephen	Chair

Example

Who ordered a printer?

```
SELECT Employees.Name
FROM Employees, Orders
WHERE Employees.Employee_ID=Orders.Employee_ID
AND Orders.Product='Printer'
```

Result

Name
Hansen, Ola

