

# Project Management Workshop

**Trainer :Hari Prasad Thapliyal**

MBA, MCA, PGDOM, PGDFM, CIC, MCITP, MCTS, CSM, PMP, PMI-ACP, PRINCE2, Agile

Project Management Mentor, Trainer & Coach

[PMI-ACP : #318050]

[PMP : #1563385]

Content related queries please write to [hari.prasad@vedavit-ps.com](mailto:hari.prasad@vedavit-ps.com)

This content is legal only for participants of VGL Jaipur May'16 Training

'PMI-ACP', 'PMI', and 'ACP' are a registered marks of the Project Management Institute, Inc.

# Trainer Introduction

- **MCA, MBA (Operations), PGDFM, PMP, MCITP, PMI-ACP, Prince2 Practitioner and CSM**
- **2 decades** of experience in IT Delivery, Training, Consulting, Agile/Scrum. **16+ years** in various IT delivery positions, **6 years** as Project Management Trainer/Coach/Mentor.
- **Trained 3000+ leaders** in India and internationally
- **190+ corporate** / open-house trainings on PMP®, Agile Project Management, PMI-ACP®, PRINCE2, PMI-RM®, PMI-SM®, CAPM®, Microsoft Project® 2007 & 2010, Enterprise Microsoft Project Server® 2007 & 2010, AgileEVM, Earn Value Management, Function Point Estimation (FPA)
- Delivered many trainings at **international location** for the citizen of **Philippines, Singapore, Indonesia, Malaysia, Thailand, Japan**
- Class room training experience of around **5500+ hours**
- FPA counting experience of more than **30000 FP** of 15+ proposals/projects
- **CMMI (1.1, 1.2), ISO9001:2000** initiative rollout
- Ex-Vice President- Volunteer Development with PMI Chennai Chapter
- Ex-Vice President- Volunteer Development with PMI Mumbai Chapter
- Ex-Chair of Ethics Committee of PMI Mumbai Chapter
- Member of PMI Mumbai Chapter & PMI Chennai Chapter
- Member of PMI (Project Management Institute) USA

# Working Agreements

- Working Time
- Break Time
- Electronics
- Corner Talk
- Group Exercises
- Participation
- Our Values (for this project)
  - Focus
  - Communication
  - Respect
  - Openness
  - Courage
  - Share



Everybody

# Workshop Agenda

## Project Planning & Management

- Project Planning
- Stakeholder Identification
- WBS (Work breakdown Structure)
- Risk Identification and Planning
- Communication Planning & Management
- Software Estimation techniques
- Scheduling and Controlling (Waterfall/Iterative /Agile)
- Resource planning & Management

# Workshop Agenda

## Requirement/ Design/ Test

- Project Management
  - Importance of requirements for project success
  - Project Management Methodology / Lifecycle Selection
  - Effective requirement gathering techniques
  - Business Requirements v/s Software Requirements
  - Verifying and Validating Requirements
  - Managing Scope
- Requirements phase in different development models (Waterfall/Iterative/Agile)
  - Requirement Elicitation techniques
  - Change Management
  - Stakeholder Management
- Design
  - Analysis vs Design in IT Projects
  - Design Methodologies
- Test Management
  - Test strategy and planning
  - Test design
  - Test case preparation
  - Test Execution
  - Defect tracking and reporting
  - Requirements Traceability Matrix
  - UAT

# Workshop Agenda

## ITIL Processes

- Production Support and Issue Management
- Release Management

# Project Management

# Project Planning

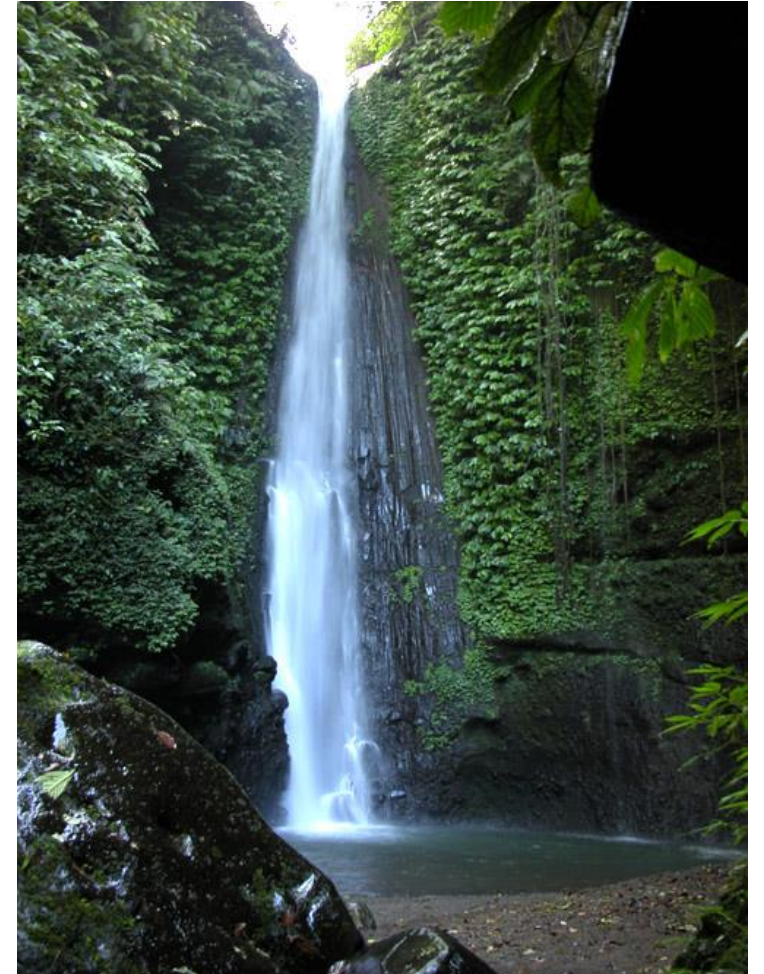
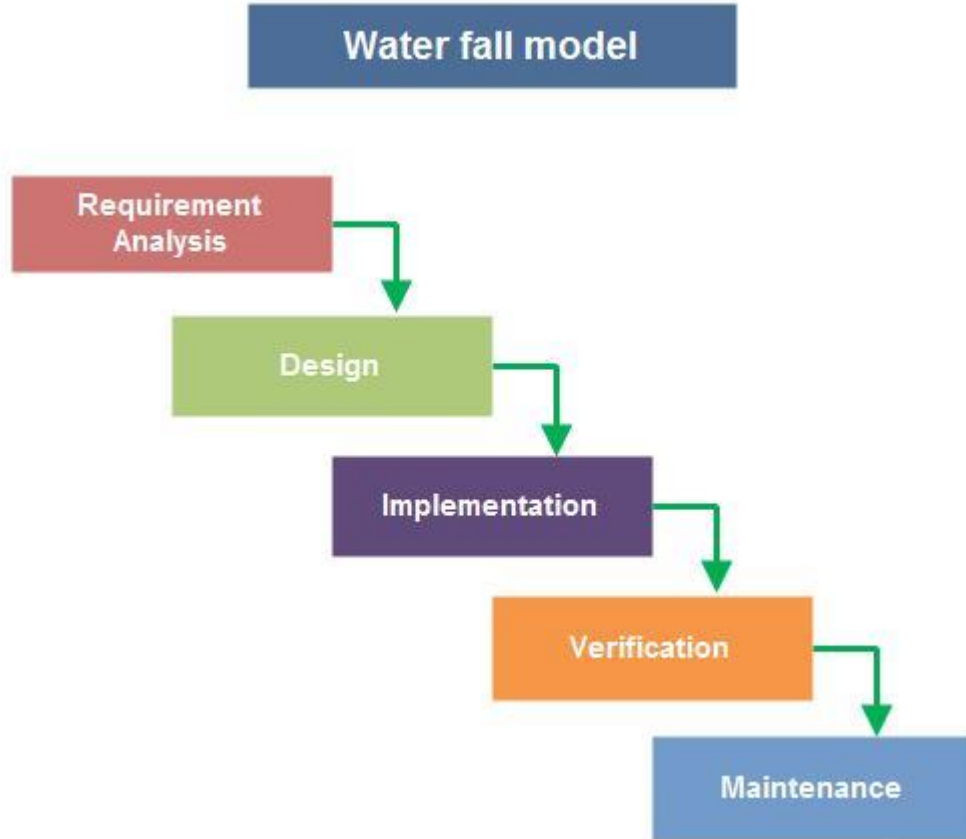
- 10 Dimension of Planning
- Component of Planning
  - How to Execute
    - What
    - Who
    - When
    - How to do
  - How to Monitor & Control (M&C)
    - What to monitor
    - How to measure
    - Who will measure
    - How to control
    - How to report



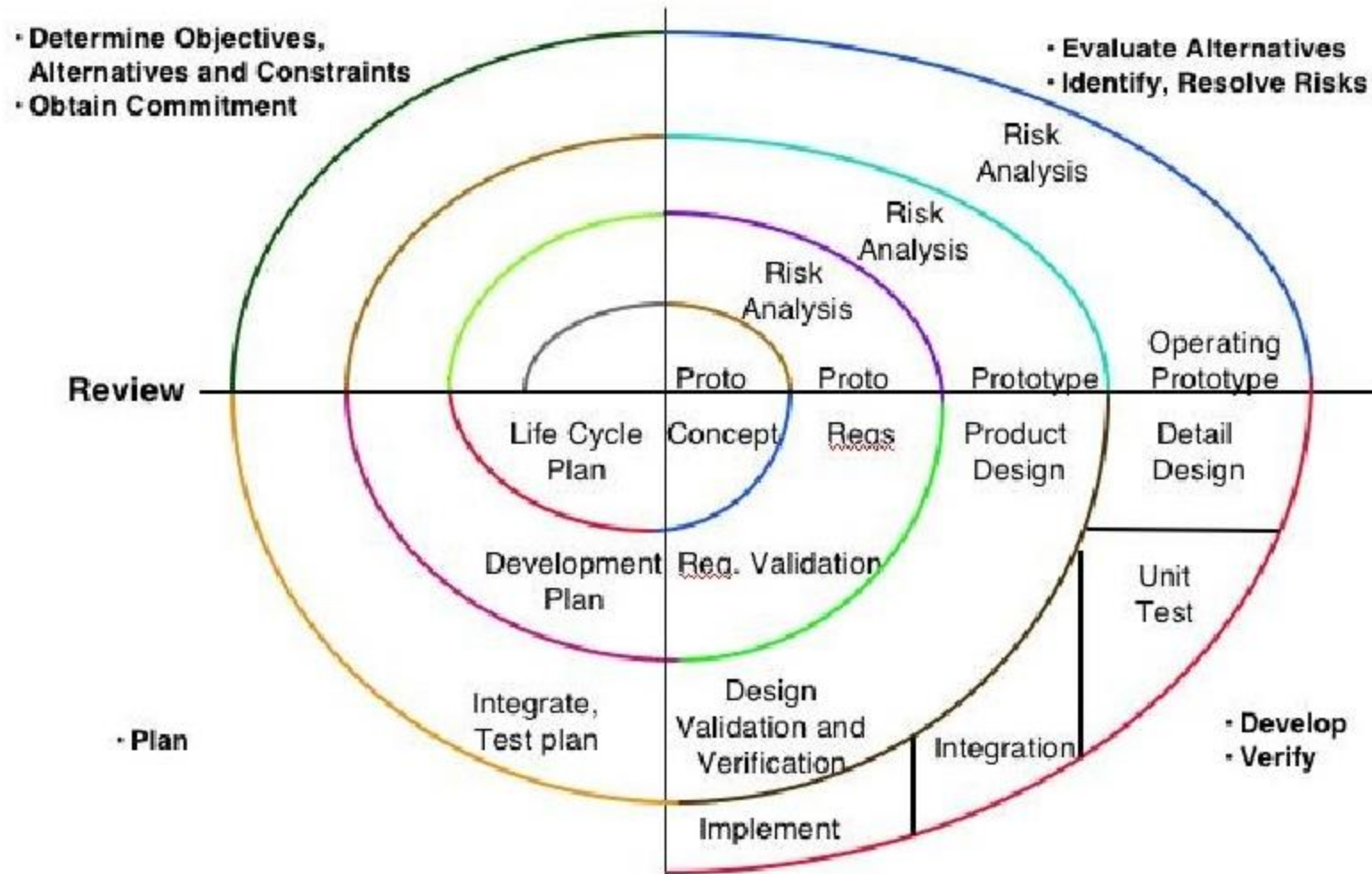
# Choose Right Methodology

- Waterfal
- Iterative
- Agile

# Waterfall

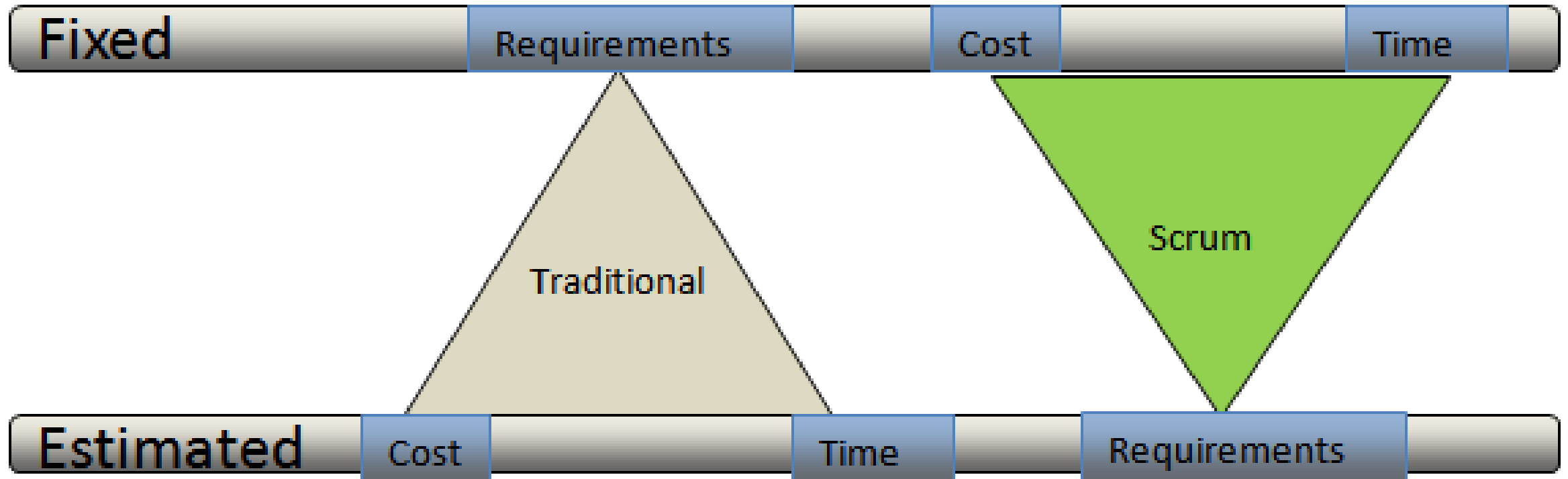


# Spiral

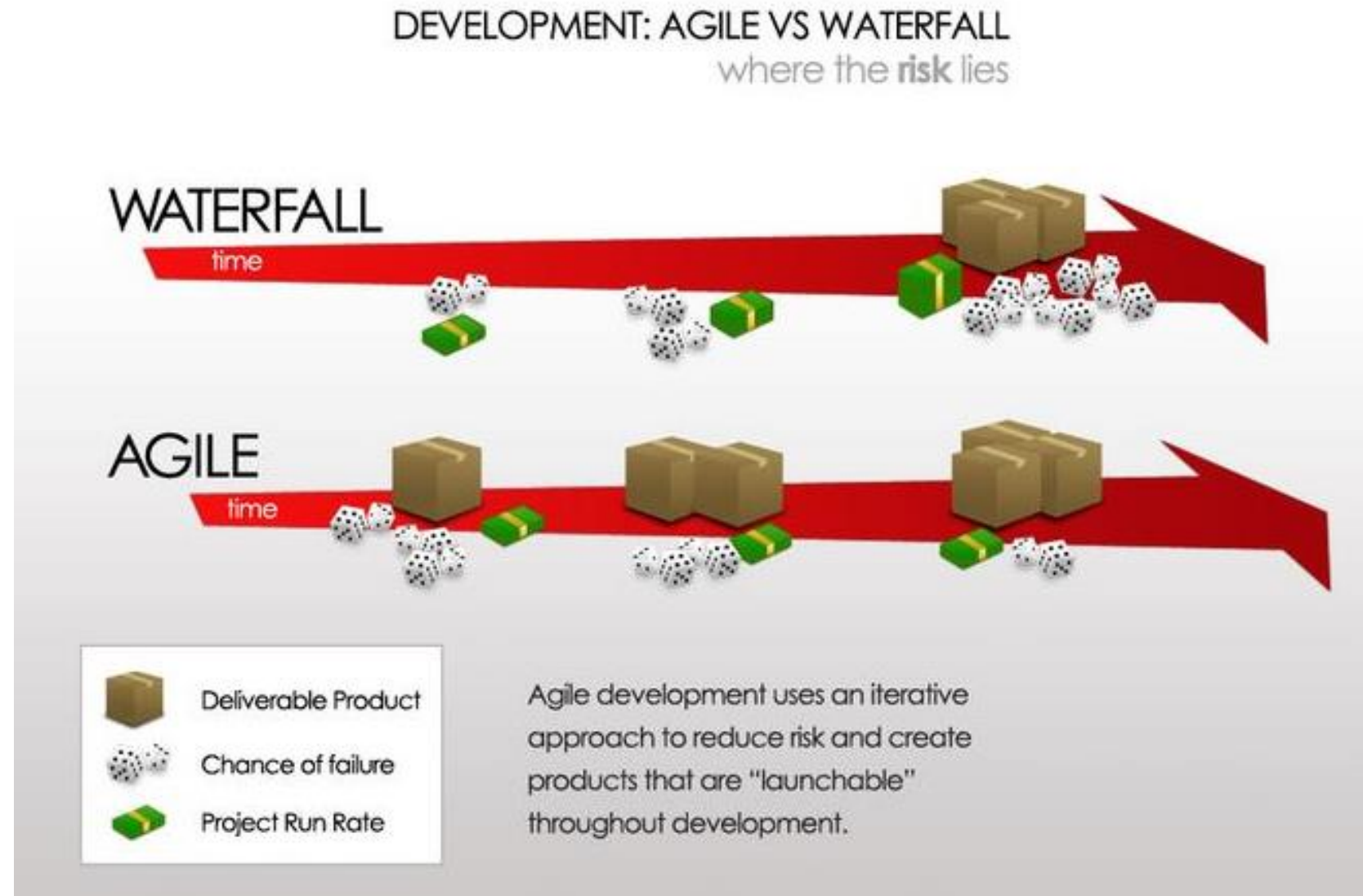


# Agile vs Traditional

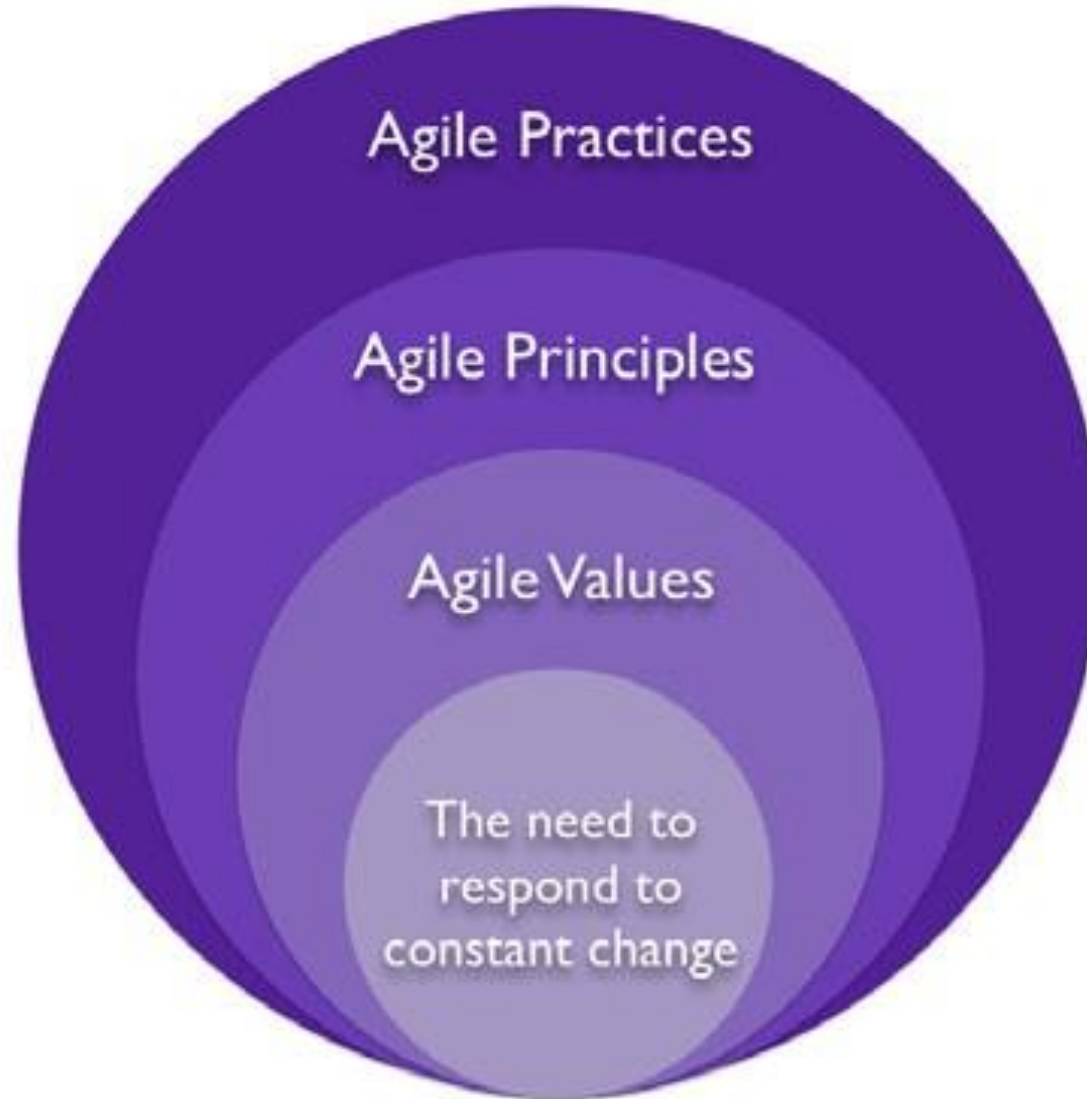
vedavitt



# Value, Risk, Productivity in Agile vs Traditional

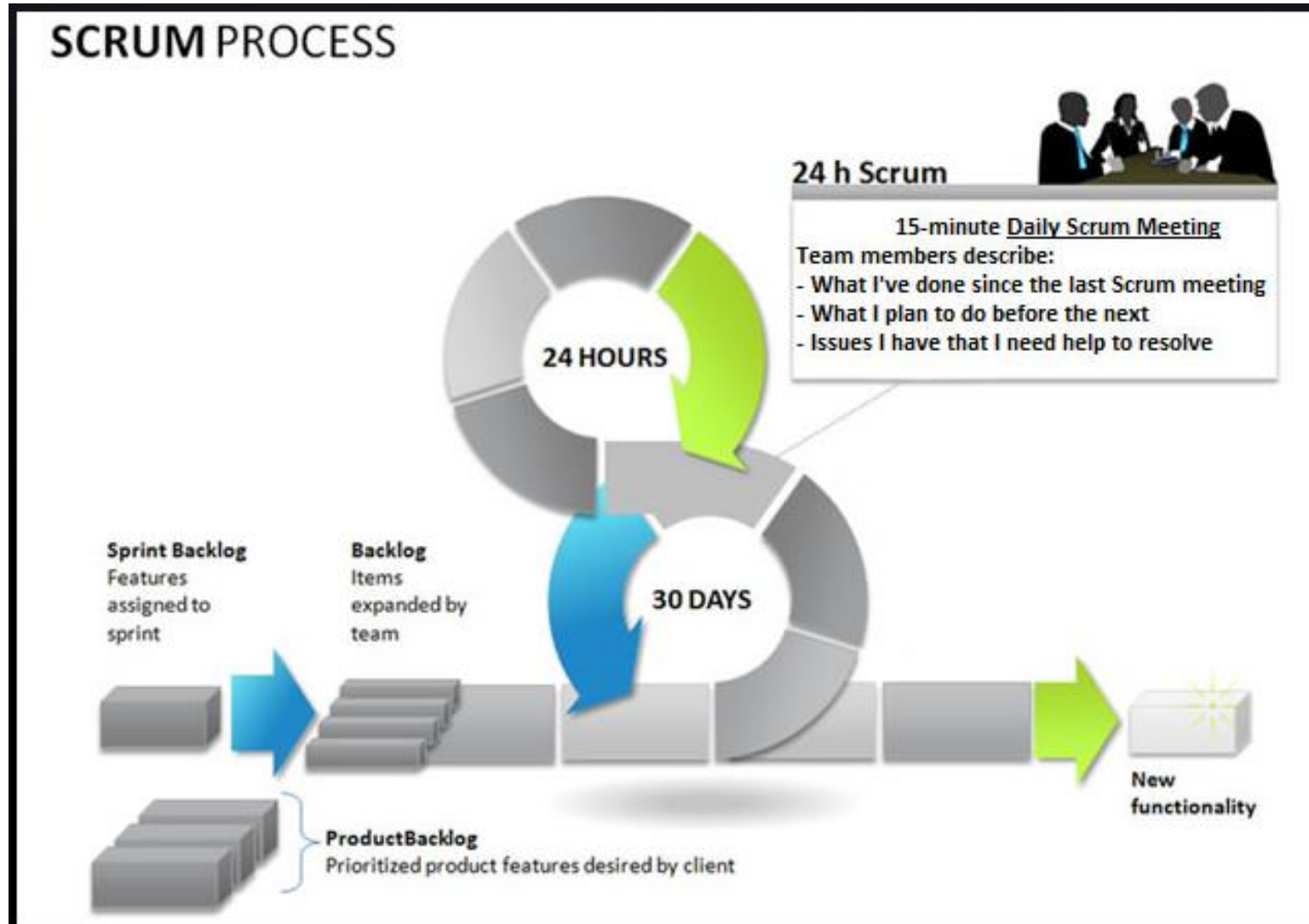


# Core of Agile





# Scrum in a Glance



# Stakeholder Identification & Management

- Who are stakeholders?
- Prepare Stakeholder Register for Every Critical Project
- Perform stakeholder interest and influence analysis
- Development stakeholder management approach for every key stakeholders
- Apply the approach
- Keep your stakeholder register updated
- Keep evaluating your approach and aligning it to the need



# WBS (Work Breakdown Structure)

- What is WBS?
- Why WBS is Important?
- How many ways of creating WBS?

# Risk Identification and Planning

- What is Risk?
- Prepare Risk Register for every critical project
- Analyse impact, probability, exposure of every risk
- Prioritize risk based on the exposure
- Understand risk tolerance of stakeholders
- Based on the exposure and tolerance make decision which risk must have response plan
- Prepare risk response plan for every high priority risk
- Every risk must have a risk owner
- Keep risk register updated
- Periodically keep evaluating your risk response plan

# Communication Planning & Management

- Every critical project should know what information need to be communicated, to whom, at what frequency, via what channel and when
- Keep your various approved reporting format ready during the project planning or modify it based on the need
- Keep communication as per the plan
- Track adhoc communication
- Evaluate your communication effectiveness on periodic basis and alter it based on the need.
- Track your communication efforts at some place

# Software Estimation Techniques

- Judgement, Experience
  - Analogous Estimation
  - 3 Point Estimation
  - Parametric Estimation
  - Story Points
- Scientific Techniques
  - Function Point
  - Use Case

# Estimation: Important Points

- No estimation is without assumptions. So document the assumptions.
- Estimation is never absolute number. So communicate estimation with range
- Estimation must be done of every important aspect of the project like size, duration, cost, efforts
- 10 Persons for 4 month = 40 person months  $\leftrightarrow$  4 Person 10 Months ☹️
- Mind the difference between calendar days, duration, efforts, ideal engineering efforts
- Even if you have to backward planning (from project end date) in a short duration project. Estimation and negotiation is must.
- That estimation is of no use which has not been communicated, agreed and baselined for tracking purpose. So communicate and get approval.
- Estimate buffer (efforts, time, cost) separately and track it separately.

# Scheduling and Controlling

- Every project manager must know the critical path and critical activities of his project.
- Critical activities are those where float / slack is zero.
- Float is not same as buffer
- Every schedule must be baselined before PM start tracking activities
- Perform plan vs actual analysis on regular basis
- Whenever deviation is beyond the threshold limit corrective and preventive action need to taken and this must be communicated to relevant stakeholders.
- Track your buffer your separetely

# Resource Planning & Management

- Every activity cannot be fastracked.
- Every activity cannot be crashed
- Resources should not be overloaded then their availability
- Productivity is never 100%
- Thinks will go wrong so plan resources accordingly so that you do not fail on your committed baselines.
- Manager buffer resources and buffer time/cost of resources carefully

# Requirements, Design & Test



# Requirements, Design & Test

- Document and track every functional and non-functional requirement through a central system
- Functional Requirement and Specification documents should have analysed requirements and system. Customer should know functional architecture of the system.
- Use difference level of abstraction diagram to visualize the functional system.
- Use difference type of architecture to visualize the solution
- Document how will you test and prove that the functional and non-functional requirements of system are completely built or partially built.
- Involve testers throughout project lifecycle. Tester and developers should not understand requirements from each other but from business analyst and customer.
- In an agile built solution and deliver value in increments

# Importance of Requirements for Project Success

- Test case for every functionality is must. This is success criteria of delivery of functionality
- Test product against the non-functional requirements
- Change management system should be agreed with stakeholders and it must be in place for fixed price project.

# Effective Requirement Gathering Techniques

- Observation & Conversation
- Interviews
- Surveys & Questionnaires
- Prototyping
- Focus Groups

\* Do not make any absolute commitment to customer about the requirement implementation till the time requirement are not analyzed and architecture solution has not committed by technical.

# Requirements Types

- Business Requirements
- Stakeholder Requirements
- System Requirements ( functional and non-functional)
- Transition Requirements

# Business Requirements

**Business Requirements** describe why the organization is undertaking the project. Project charter, business case, or in a project vision and scope statement. You can't build software from such high-level information. It help get the project owner, stakeholders and project team on the same song sheet.

- Problem Statement
- Project Vision
- Project Constraints (Budget, Schedule, and Resource)
- Business Objectives
- Project Scope Statements (Features)
- Business Process Analysis
- Stakeholder Analysis
- IT Service Impact Analysis

Ex: "We need to establish an online customer portal." "The portal should list our products under various categories."

# Stakeholder Requirements (Software Req.)

- **Stakeholder Requirements**, user needs, user requirements, describe what users do with the system, such as the activities that users must be able to perform. These are generally documented using narrative text, use cases, scenarios, user stories, or event-response tables. Generally documented in a User Requirements Document (URD). Signed off by user community.
- Ex: Use Case for Login
  - 1. Go to website
  - 2. Click on login
  - 3. Enter username and password
  - 4. You are redirected to the start page.

# Solution Requirements (Software Req.)

- **Solution (System) Requirements** are what the developers use to build the system. These are the traditional “shall” statements that describe what the system “shall do.” System requirements are classified as either functional or nonfunctional requirements.
- A **functional** requirement specifies something that the developer needs to build to deliver the solution.
- Ex: "The system shall be able to register a product using the following fields: Name (20 characters long), Details (2000 characters long), Price (currency), Category (pick list)."
- Ex: "The system shall support that up to 5 pictures can be listed per product."

# Non-Functional Requirements

**Nonfunctional** requirements specify all the remaining requirements not covered by functional requirements.

Ex: “System should be able to support 50 concurrent users” “Interest and due calculation report should be ready with 3 seconds” “Application should work on all latest browsers”

- Business continuity, Compliance, Interoperability, Maintainability, Performance
- Interface: Integration with other system
- Operations: Installation, system management
- Reliability: Failure rates, availability, recoverability
- Usability: Different ability people, navigation, help, documentation, difference experience level people
- Legal, packaging, hardware



# Transition Requirements (Software Req.)

**Transition Requirements** describe capabilities that the solution must have in order to facilitate transition from the current state of the enterprise to a desired future state, but that will not be needed once that transition is complete. They are differentiated from other requirements types because they are always temporary in nature and because they cannot be developed until both an existing and new solution are defined. They typically cover data conversion from existing systems, skill gaps that must be addressed, and other related changes to reach the desired future state. They are developed and defined through solution assessment and validation.”

# Verifying and Validating Requirements

- **Verification** is the process of confirming that the designed and built product fully addresses documented requirements. Verification consists of performing various inspections, tests, and analyses throughout the product lifecycle to ensure that the design, iterations, and the finished product fully addresses the requirements.
- **Validation** is the process of confirming the completeness and correctness of requirements. Validation also ensures that the requirements: 1) achieve stated business objectives, 2) meet the needs of stakeholders, and 3) are clear and understood by the developers. Validation is essential to identification of missing requirements and to ensure that the requirements meet certain quality characteristics.

# Managing Scope & Change Management

- Define change control board and their threshold limits.
- Define process of managing changes and make sure team follow that
- Understand the meaning of scope creep and manage it
- Have project boundary defined in terms of within scope and out of scope in business requirement document.

# Requirements Phase in Different Development Models

- Waterfall : The first phase of project lifecycle
- Iterative : At the start of every iteration. Output of iteration may be architecture, non-working prototype, POC
- Agile : Requirements are elaborated during product backlog grooming in every sprint for the future sprint work.

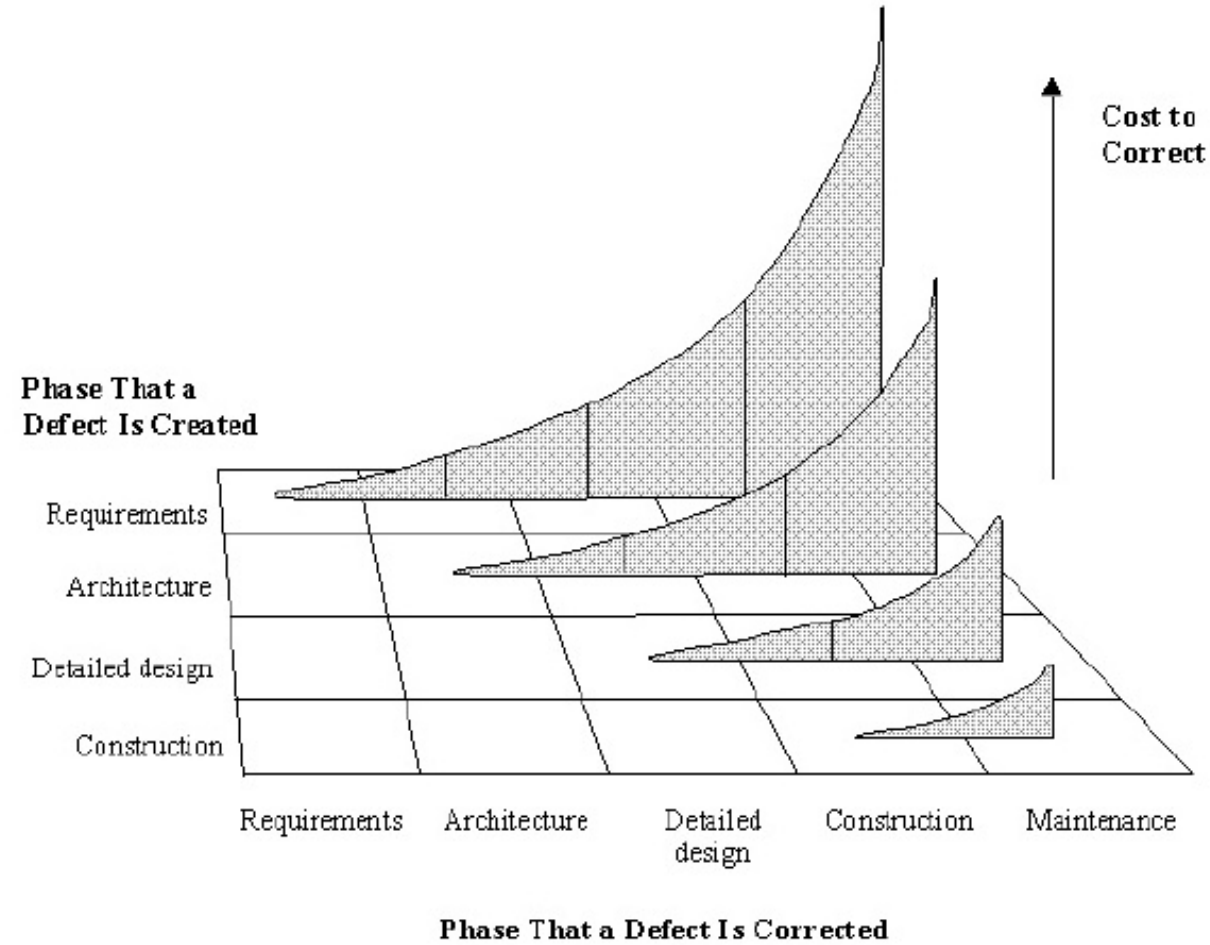
# Requirement Elicitation Techniques

- Reference document, product
- Para-phrasing
- Throwaway prototype (paper, board, electronic)
- Speak user's language

# Analysis vs Design in IT Projects

- Analysis
  - Business Rule
  - Workflow
  - Duplicity
  - Priority
  - Usecase
  - Consistency

# Why are Requirements so Important



Copyright 1998 Steven C. McConnell. Reprinted with permission from *Software Project Survival Guide* (Microsoft Press, 1998).

# Integration & Integration Testing

- No last-minute change request.
- Defect is not change request.
- Ensure defect fix do not break other part of the system (automation)
- Automate as much as possible to guarantee high quality
- Involve customer



# Release Management

- Release calendar
- Release plan (hardware/software, responsibilities, activities, validation, plan B)
- Migration Strategy
- Minimal Downtime
- Configuration Control
- Release Documents
- Committed Team
- Defect Prioritization

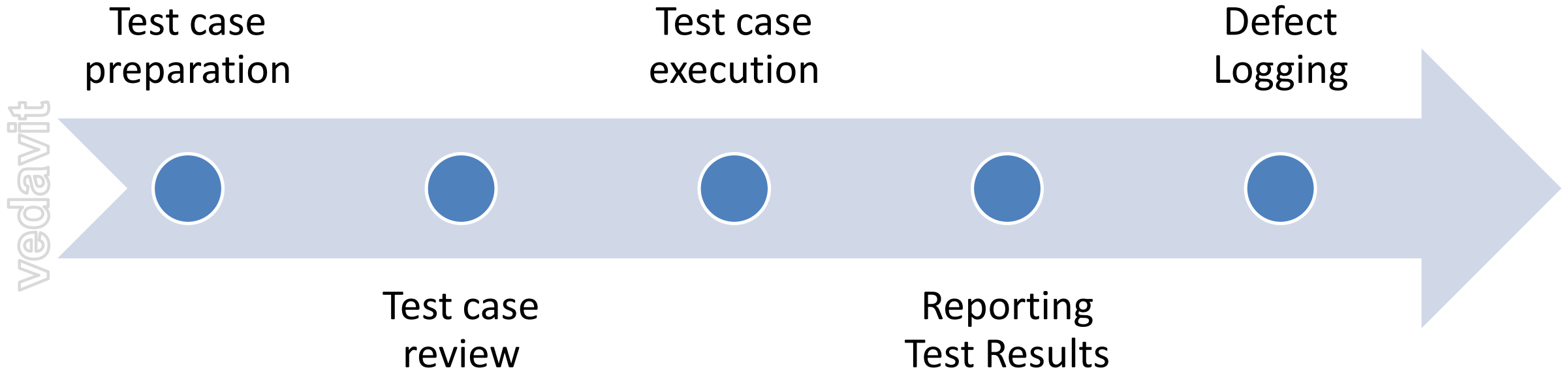
# Design Methodologies

- **Hierarchy Architecture / Structured Analysis and Design Technique:** This approach utilizes a diagram to describe the hierarchy of a system's functions.
- **Data Structured Systems Development:** Data structure determines the system structure in this methodology.
- **Object Oriented Design:** This methodology is based on a system of interacting objects.
- **Agile Design**
- **Heterogeneous Architecture**
- **SOA Architecture**
- **Interaction Oriented Software Architecture**
- **Master/Slaves Software Architecture**
- **Data Flow Architecture**
- **Top Down Design or Stepwise Refinement:** This starts from the end solution and works backwards, refining each step along the way.
- **Bottom Up Design:** This methodology starts with a foundation and works up towards a solution.
- **Structured Design:** This is an industry standard. The technique starts by identifying inputs and desired outputs to create a graphical representation.

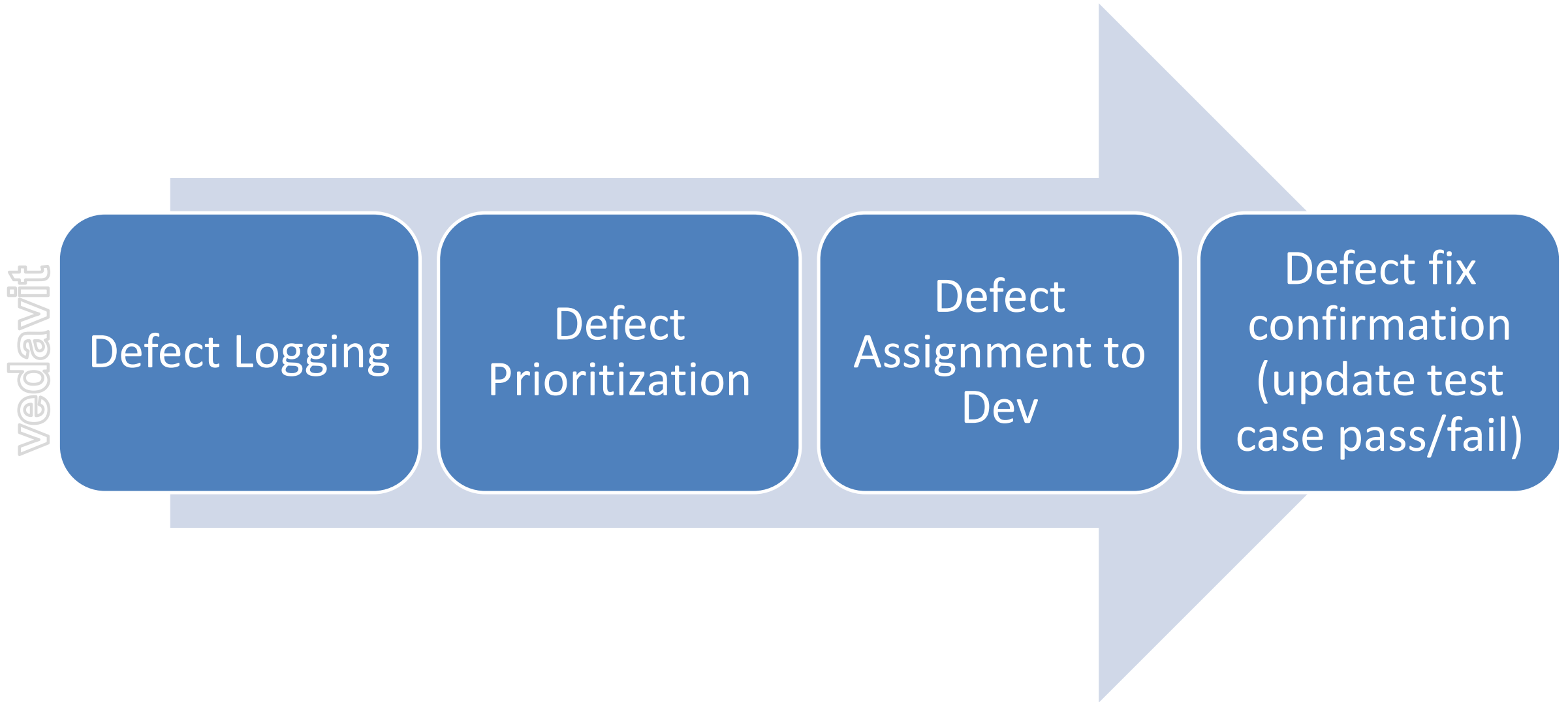
# Test Design /Test Strategy and Planning

- Based on the complexity of application make what different level of testing will be performed.
- Make a plan to ensure each type of testing. (Infrastructure, Responsibilities, Activities, Schedule, Cost, Test Case, Reporting, Defect Fixing)
- Is any third party certification required after testing

# Processes for Testing & reporting



# Defect Tracking and Reporting



# Requirements Traceability Matrix

- Use RTM to track the progress of each functionality
- RTM is excel sheet kind of column structure
- RTM contains: Req Id, BC Ref, Design Ref, TC Ref, Dev Ref, Delivery Status, Rel. #, Owner
- Update RTM during and verification and validation
- Use RTM for preparing delivery status report

# UAT

- Help customer in writing User Acceptance Testcases in advance
- Encourage customer to test the product using UA Testcases
- Define UAT environment and validate before UAT start
- Train customer about how/where to log UAT defects
- Dedicated team to fix the UAT defects on priority and made decision
- Do not work on any new requirements while fixing any UAT defect

# ITIL Processes



# IT Systems Includes

- Firewalls
- Various type of local, intranet and cloud Storage
- Directory Services
- Various Types of Database Servers
- Various Types of OS
- Various Types of Web Client
- DHCP Services
- Application Servers
- Various type of Network devices and protocols

# Challenges

- Change in business happening every moment and this sends out it ripples
- Challenges is how to ensure high security, performance, availability with distributed and heterogeneous system and manage all this with lesser cost and outsourced team!

# So..

- How they manage the complex challenges
  - Banks,
  - Insurance
  - FMCG
  - Transporters
  - Airlines
  - Financial Security
  - Defence
  - Oil & Gas
  - Universities

# Learn from others...

- Processes
- Functions
- Roles

That how they manage services and service lifecycle

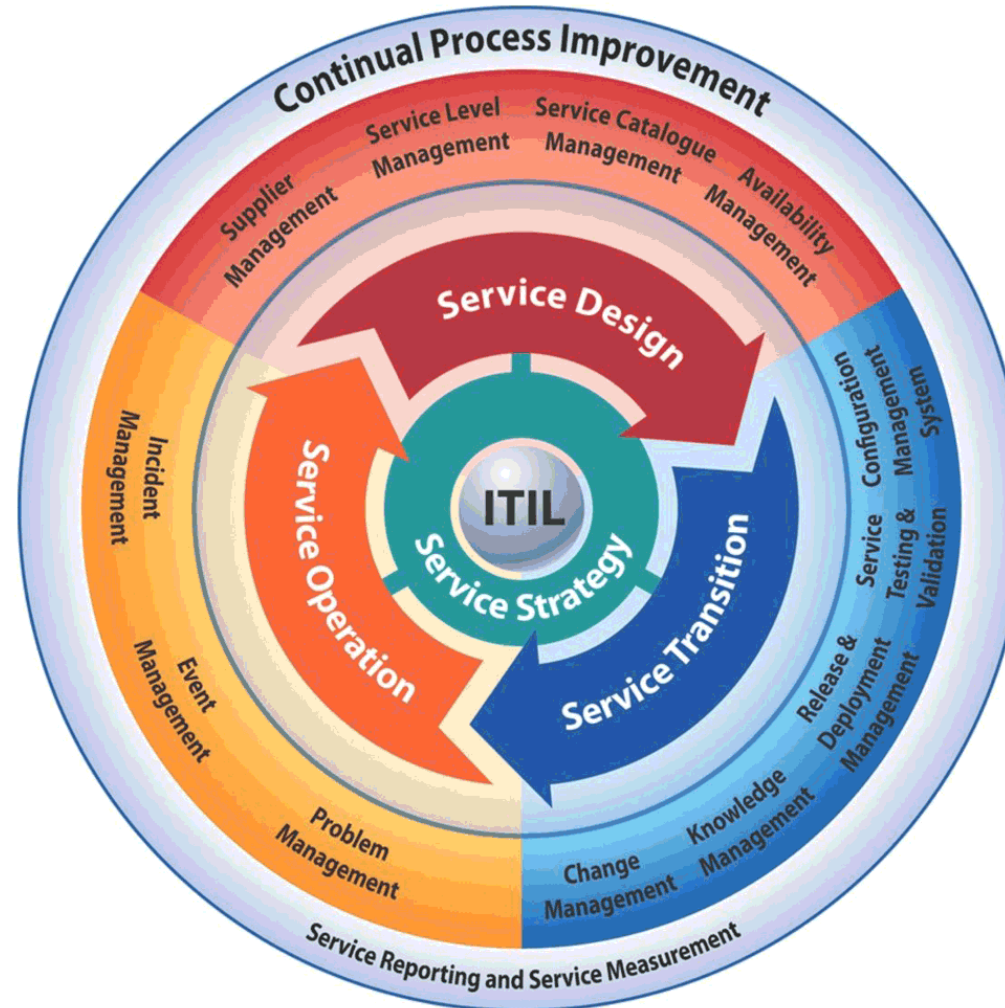
# What is service?

- The best way to think about a service is to think terms of customer. Why should they pay for it? Business is the customer of IT.
- If services are not available then it's not a service. So availability management is the key.
- All the service support and service delivery disciplines work together to deliver agreed levels of service availability to our customers.

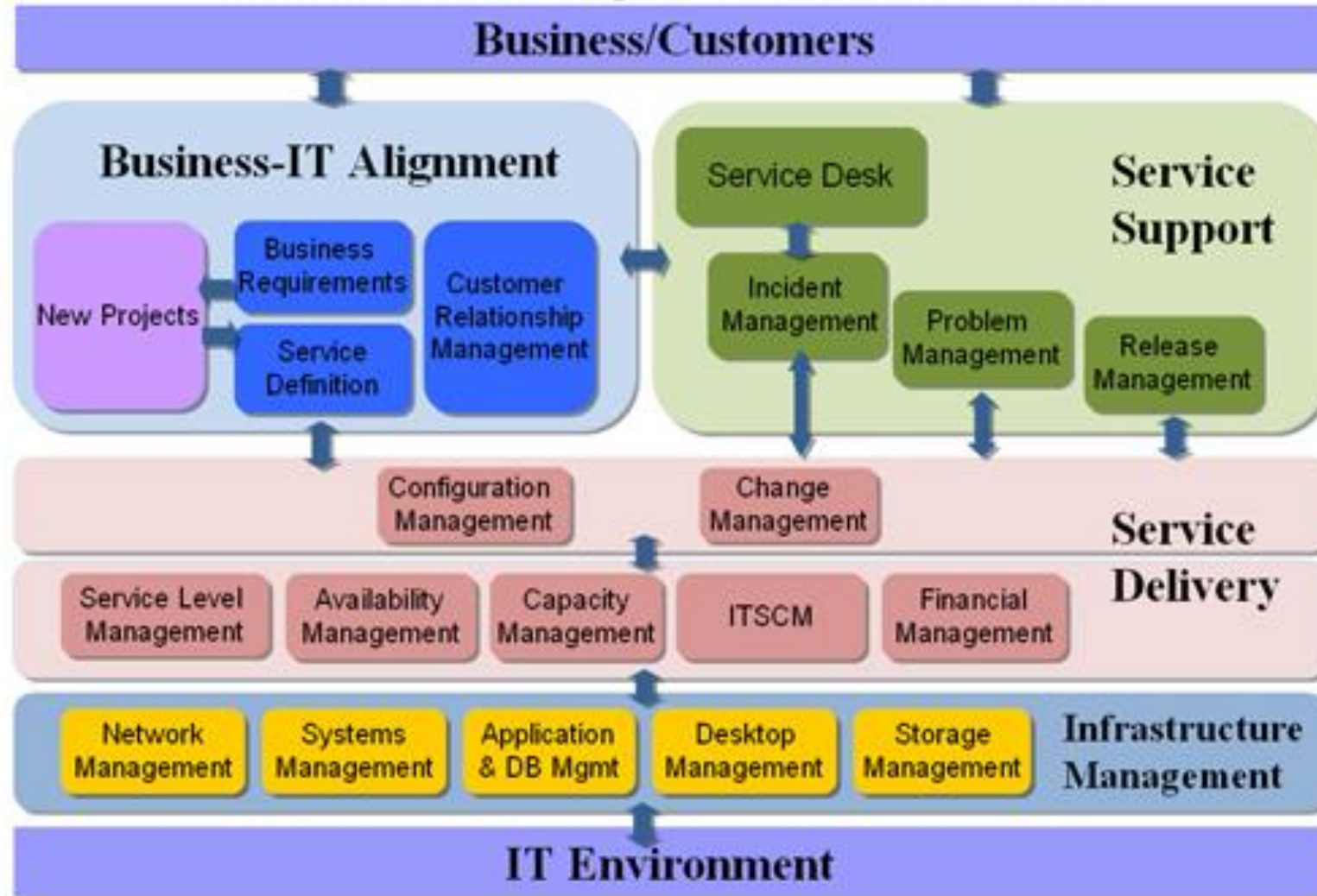
# Functions and Roles

- 4 Functions
  - The Service Desk
  - Technical Management,
  - Application Management,
  - IT Operations Management
- Roles & RACI
  - Process owner
  - Service owner
- Technology and Architecture for Service Management & Service Automation

# ITL Core Processes



# ITSM Management Process





# Processes

IT Service Strategy	IT Service Design	IT Service Transition	IT Service Operation	IT Continual Service Improvement
<ul style="list-style-type: none"> <li>1. Portfolio Management</li> <li>2. Financial Management</li> <li>3. Demand Management</li> </ul>	<ul style="list-style-type: none"> <li>1. Service Catalogue Management</li> <li>2. Service Level Management</li> <li>3. Capacity Management</li> <li>4. Availability Management</li> <li>5. Continuity Management</li> <li>6. Information Security Management</li> <li>7. Supplier Management</li> </ul>	<ul style="list-style-type: none"> <li>1. Support &amp; Transition Management</li> <li>2. Change Management</li> <li>3. Asset &amp; Configuration Management</li> <li>4. Release &amp; Deploy Management</li> <li>5. Validation Management</li> <li>6. Evaluation Management</li> <li>7. Knowledge Management</li> </ul>	<ul style="list-style-type: none"> <li>1. Event Management</li> <li>2. Incident Management</li> <li>3. Problem Management</li> <li>4. Fulfillment Management</li> <li>5. Access Management</li> <li>6. Service Desk Function Management</li> <li>7. Service Operations Function Management</li> <li>8. Technical Operations Function Management</li> <li>9. Application Operations Function Management</li> </ul>	<ul style="list-style-type: none"> <li>1. IT Governance Management (using COBIT best practices)</li> <li>2. IT Resource Management</li> <li>3. IT Quality Management (using Six Sigma methods)</li> <li>4. IT Security Management (using ISO standards)</li> <li>5. Service Measurement</li> <li>6. Service Reporting</li> <li>7. Service Improvement</li> </ul>

# Products (Tools)

IT Service Strategy	IT Service Design	IT Service Transition	IT Service Operation	IT Continual Service Improvement
<ol style="list-style-type: none"> <li>1. Service Request &amp; Planning Tools</li> <li>2. Service Knowledge &amp; Configuration Management Tools (SK-CMS)</li> </ol>	<ol style="list-style-type: none"> <li>1. Service Catalogue Tools</li> <li>2. Service Level Management Tools</li> <li>3. Capacity Planning Tools</li> <li>4. Service Modeling Tools</li> <li>5. SK-CMS</li> </ol>	<ol style="list-style-type: none"> <li>1. Asset Management Tool</li> <li>2. Service provision Tool</li> <li>3. Run Book Task Automation Tools</li> <li>4. SK-CMS</li> </ol>	<ol style="list-style-type: none"> <li>1. Service Desk with Incident Management Tool</li> <li>2. Problem Management Tool</li> <li>3. Event Management Tool</li> <li>4. Run Book Technology Troubleshooting Tool</li> <li>5. Run Book Application Troubleshooting Tool</li> <li>6. SK-CMS</li> </ol>	<ol style="list-style-type: none"> <li>1. Compliance Management &amp; Measurement Tools</li> <li>2. SK-CMS</li> </ol>

# People

IT Service Strategy	IT Service Design	IT Service Transition	IT Service Operation	IT Continual Service Improvement
1. Service Definition Manager 2. Service Research Manager 3. Financial Analysis Manager 4. Service Marketing Manager 5. Service Forecast Manager	Engineering Manager for 1. Security 2. Desktop 3. Network 4. Systems, Servers & Storage 5. Applications	Assets Manager for 1. Security 2. Desktop 3. Network 4. Systems, Servers & Storage 5. Applications	Operation Manager for 1. Security 2. Desktop 3. Network 4. Systems, Servers & Storage 5. Applications	1. Service Measurement Manager 2. Quality Measurement Manager 3. Compliance Measurement Manager 4. Security Measurement Manager 5. Resource Measurement Manager

# Discussions !