System Architect Essentials

Principles of Application Development

# Introduction to principles of application development

In this lesson, you will learn 5 key principles of application development.

## Objectives

At the end of this lesson, you should be able to:

* State the importance of using **Pega’s Directly Capture Objectives™** approach to managing requirements.
* State the benefits of using **Pega’s Situational Layer Cake™** architecture to design an application.
* State the benefits of using a model-driven application design.
* State the benefits of using a responsive user interface design.
* State the benefits of using Pega’s data management capabilities.

# Capture objectives directly in the application

* Business application development teams can find it difficult to communicate business requirements.
* There may be no common language between the business and IT stakeholders. And, there may not be a common view of the business goals.
* Often, business stakeholders are not sure of what their business needs are.
* When this happens, IT stakeholders find it difficult to get the details they need.
* Business and IT stakeholders must share a common understanding of the business requirements.
* You also need a way to ensure business requirements are current and available to all stakeholders.
* In Pega 7, **you capture business requirements directly in the application**. This practice is called **Directly Capture Objectives,** or **DCO.**
* Video
* DCO ensures business and IT stakeholders share a common understanding of the business requirements.
* DCO also ensures the business requirements are up-to-date and available to everyone.

# Build multi-dimensional applications

* The critical dimensions of any business are product, region, channel, and customer.
* When you conduct business in different countries, you must manage the regulations of each jurisdiction, and the cultural differences in each region.
* When you sell multiple products through multiple channels, you must manage the business rules for selling each product in each channel separately.
* When you sell to different types of customers, you must manage each customer’s expectations and preferences.
* With some application development platforms, you must create separate copies of the application for each product, region, or channel.
* Or, you must create an application that treats all business transactions the same, regardless of the business context.
* The result is enterprise applications that are hard to maintain, and even harder to change.
* Pega uses a **unique application architecture** called a **situational layer cake.**
* The Situational Layer Cake allows you to organize your application using the same dimensions as your business.
* The situational layer cake makes reusing common policies and procedures easy while allowing for differences between products, regions, channels, and customer segments.
* Video
* Pega’s unique approach to enterprise application architecture – the Situational Layer Cake – can help turn the complexity of an ordinary enterprise application in a simple and coherent end-to-end customer experience.

# Use a model-driven application design

* Ask any business person to explain their needs for an enterprise businesses application.
* As they explain their needs, you will notice they do not dive into the details about any particular part of their application.
* And, they do not discuss the behind-the-scene technologies needed to make the business application useful.
* When business people explain their needs for an enterprise business application, they describe the major steps of how work gets done.
* They talk in terms of a case and a desired outcome, and the stages that case may go through until the desired outcome is achieved.
* What they describe is the life cycle of a case.
* To be effective, your application design and development efforts must match the way business people naturally talk about their work.
* A case, and it’s life cycle, is the central methapor in Pega’s model-driven approach to building business applications.
* Watch this video to explore Pega’s approach to case life cycle management.
* Video
* Rather than drawing a complex end-to-end diagrams, Pega 7 allows you to build a visual representation of the life cycle of a case.
  + Essentially building the skeleton on which you hang the more detailed processes.
  + This allows you to establish a business view of the case before debating the details.

## GLOSSARY

* requirements
  + A requirement is an event, condition, or function that must be satisfied and tracked by an application.
  + A requirement can be functional (a capability) or non-functional (a quality or constraint).
  + One ore more requirements define the criteria for the successful implementation of a specification.
* situational layer cake
  + A situational layer cake is a metaphor to describe how Pega interacts with customers the right way at the right time.
  + Each layer specializes on a different dynamic variable, such as customer attributes; country or region; product or line of business.