**PEGA Project Implementation & Methodologies**

**- Requirements**

**- Analysis**

**- Prepare Test Case Documents**

**- New/Existing Application ( Scratch/Proposed Functionalities)**

**- What are the Current Functionality &  What will be the Future Functionality. Prepare the Test Case Documents. Share these documents with TechLeads & BA**

**- BA will Prepare the Requirements**

**- BA will have complete idea about application functionality ( End users also have Functional knowledge)**

**- BA will identify No of Requirements  ( 10), Duration ( 3 Months) , Decide the Release date. ( When the code will move to the Live Server)**

**Methodology:**

**Water Fall Methodology:**

**- Requirements**

**- No Deadlines ( Dead Lines can be extended)**

**- Status Calls, but specific process to set the call**

**- Requirement- You will do analysis, No Doc Preparation, For Proposed functionalities no doc required.**

**- You can attend Status & discuss everything, application walkthrough is not mandatory.**

**- Initially we start with waterfall to gather the requirements**

**- To Post Requirements they can use Jira Tool'**

**- They will open request & write down the Requirements**

**- Attach requirement documents( Word, Excel, Images)**

**- Now they will upload all requirements to jira tool**

**- After that they will notify the Tech leads**

**- Tech Leads will login - > Gothrough the Requirements->**

**- Every Requirement will have Requirement ID ( ABCReq01)**

**R1: Use Case/ Requirement**

**R1.1 : User Stories/Requirement Points**

**R1.2**

**R1.3**

**10 Requirements = 10 Use Cases**

**R1.1 -> User Stories = Requirement Points**

**Agile /Scrum Methodology:**

**-Scrum is Part of  Agile Methodology**

**-It is a Defined Structure**

**- Scrum Masters ( Tech Managers)**

**- These scrum masters can be managers from ur company or from the vendor company.**

**- Scrum masters will decide giving points to the requirements**

**- We will have User Stories**

**- For each story they will give one story point. In scrum we have story points**

**- 1 Point= 1 day/2 Day**

**- We have to finish the task within the given point**

**- If u finish with in given points , they will revisit the requirement & reduce the point. Duration & efforts will be reduced, the cost to vendor also reduced**

**- In scrum we have daily status calls within ur team, & with ur BA.**

**- we will never communicate with business users directly ( Until itis a support Project)**

**- During status calls there will be teams from diff locations, BA also will join. BA can be from same team or from client team also.**

**- In status call we can discuss about the work progress, any dependency issues, doubts.**

**- In scrum we will never miss the deadlines**

**- Ur first point of contact will be ur team lead. Without informing TL/Lead Architect we should never go to BA**

**- Fixed Dead Lines ( Sprints)**

**- 1 Sprint = 2 weeks ( 10 Days)**

**- In Sprint -> we have Use cases ( 10 Req) - 5 Dev**

**- Each  developer will take 2 requirement  & Complete with in 1 Sprint (2 weeks)**

**- Once Development is Completed, U will do Unit Testing**

**- Unit Testing- ( Complete all teh Test Cases)- Positive & Negative Test Cases & Verify all**

**InFlight: The WO Created before Production**

**FutureCase: The WO Create after Production**

**- We need to Provide Unit Test Results ( WO Id's)**

**- Once Unit Testing is Completed , Notify the Tech Leads**

**- Tech Leads will create the Product Rule ( Packaging) - > From Product Rule They will create Product File.**

**DevEnv---> QA/Testing Env ----> UAT ---> Live Env**

**Testing:**

**Testers will verify all the test cases. They will prepare their own test cases.**

**- If they find any fault then they will raise a Defect.**

**Defect: DMS ( Defect Management System)/ Quality Center**

**- They will raise a Defect, and upload to the Tool**

**- They will generate the Defect ID ( Step by Step Logs & Description)**

**- What is functionality missing & What has to Correct. What error & also add error screen shots.**

**- Developer will login to Defect management System, understand & work on it.**

**- Once Verify & Create a Product & Move to QA Server.**

**- QA will do the Testing, it is working fine means they will inform BA,  QA Manager will approve It. Then Code can be moved to UAT Env.**

**UAT Env/Pre Production: User Acceptance  Testing ( Regression Testing)**

**They will perform Regression Testing ( Complete Testing)**

**- Testing will be done for all the test cases. old Cases, & everything they will test.**

**- BA will also do some testing in UAT Server**

**- Regression Testing is Completed they will move the code to Live Env.**

**Production Env**

**- They will deploy on the Release date.**

**- As a Developer before release date , we have raise the ticket.**

**Ticketing Tools: Service Now or Service Center**

**- In this Ticket we have to include, what is the application, Product release date, what are the modules & approvers ( BA, Manager, Business Manager).**

**In Ticket they will also attach implementation plan ( These are the instructions for release team)**

**- All Above must approve, then only code will be moved to Production**

**- without implementation plan no code will be moved in any technology.**

**Release Team:**

**Release team in separate, they will do the production deployment**

**- Release Management Team:**

**They will go ahead & get out request. They will take the code & Deploy to target env.**

**- Deployment will bot happen through pega tool.**

**- They will use some specific third party tools: ( ARM, GIT, SVN)**

**- These tools will copy the code in to respective folders & Scripts will be running & complete all the setup & configuration.**

**The Tools will follow the Instructions:**

**- What code to Move**

**- Backup Plan ( Back up Tools)**

**- If anything failed, they will implement back up plan & redeploy the Code**

**- Development team & Testing team will be available during deployment for support.**

**- Sigh off**

**- BA will do testing again with available Test cases in Live Env.**

**Tools Used in PEGA**

**Requirement Gathering Tools: Jira, lotus , rally, poker**

**QA: DMS Tools-  Jira, Service Now, ALM**

**Release management: GIT, ARM, SVM**

**Deployment: PEGA PRPC, ARM**

**Team Size: Tech lead, BA, Team lead,  Developers**

**Integration Testing: SOAP UI ( SOAP) , POST MAN (REST)**

**====**

**Login Types:**

**Developer Login: Development url   ( Design & Complete the Testing)**

**Single Sign on/LDAP:  Plain Screens, u will login with user Name & password. ( Again u will do testing in SSO, If testing is Succesful in SSO then u have tested exactly how end users access in the live Env). Then we can conclude the code is working fine.**

**=================**

**What Kind of Project to Choose?**

**You can choose any domain Expect Social Networking/Retails Applications.**

**- While Explaining any project be aware of the business.**

**- Learn from the Google**

**Ex:  Top 50 health care applications in USA**

**- What is input & Output of the Project ( About )**

**- We need to identify technical terms through Google. Include some technical terms in ur project**

**- Make a Note of all the screens in the Application**

**- Think about every screen from ur application**

**- Implement Every Rule from your pega rules.**

**Ex:**

**Quote:**

**Policy No**

**Effective Data**

**Policy End Date**

**Premium Calculation/Amount**

**Coverage's- Deceases Covered**

**Risk**

**Additional Coverage's**

**To Explain any Project:**

**My Project is Insurance Project, Which is being used by the Back operations Team. Agents will collect details from the customer.**

**Back operations team opens application and submit all the details. After Submission they will generate Policy No with quote Premium.**

**The Policy no will be Generated with Final Premium & Policy Term**

**Like This any project in a general way u can explain.**

**- Identify 1 task for every rule within the application**

**====**

**Any type of application the design will be similar, The Terms will change.**

**Insurance:**

**Personal Insurance**

**Health Insurance**

**Home Insurance**

**Life Insurance**

**=**

**What is ur Client emailid:**

**U Can say, I can't provide my Client EmailId, I will provide my Parent Company emailID.**

**--**

**Why did you complete the Certification Recently?**

**- I'm working as budding Resouce in the Project & recently billing in done.**

**S0, I completed my Certification recently.**