Atik Howlader

Course Notes

8/2/2020

Software Development Life Cycle (SDLC)

* To build a software from when you think about it to when you develop it
* A step by step process involved in the development of a software product
* A set of phased processes that guides a systems development effort from its inception through its implementation

SDLC Phase

* Initial Phases (Requirements)
* Analysis Phase (Analyze details)
* Design Phase
* Coding Phase
* Testing Phase
* Delivery and Maintenance Phase (Deployment)

Initial Phases

* A concept – Feasibility study, Demand Assessment, RDI Matrix, Estimation
* Kick Off Meeting- A Project Manager is placed in charge and he/she will develop a Concept Proposal/SOW
* A high-level view of the intended project and defining it’s goals
* The Engagement Manager (EM) would discuss the Financial Matters
* Service Level Agreement (SLA) – services, support, cost….
* Project Approval Document (PAD) – Formal approval by client
* The Business Analyst (BA) will gather information/ client requests and initiate documentations
  + BDD – Business Development Design
  + BRS – Business Requirement Specifications
  + FRS – Functional Requirements Specification
* Output – Feasibility study, outline the business goals, define the project’s scope, assess project demand

\* SOW – Statement of Work

Analysis Phase

* The Project Manager prepares the Project Plan/Charter, PID
  + Three dimensions/triple constraints of PP – Resources, Time, Money
* Business Requirements – The BDD/BRD document is taken as the input
* Output document for the phase is Software Requirements Specification (SRS), Functional Requirement Documents (FRD), Use Case or User Story – prepared by System Analyst (SA)/ Business Analyst (BA)/ Subject Matter Expert (SME)
* Work Breakdown Structure (WBS) – Tasks to be completed by teams

\*Project Plan is to layout the project

Design Phase

* High Level Designing – In this level of Designing the project is divided into number of modules – Technical Manager (TM) or Chief Architecture (CA) or Solutions Architect (SA)
* Low-Level Designing – In this level of Designing the modules are further divided into the number of submodules – Dev Team Lead (DTL)
* The Chief Architect/Developer’s Lead would prepare the Technical Design Document (TDD) or Technical Requirement Documents (TRD), or Detail Design Documentation (DDD)
* Take those Requirements and develop detailed, workable specifications
  + Everything is put together and the actual design of the system is done
  + Documentation such as the Maintenance Manual, Operations Manual, and training manual begin in parallel

\*All requirements should be in place after the Design Phase

Coding Phase

* Developers would write the Programs for the Project by following the coding standards & Project Documentations, Use Case, User Stories, TRD, TDD, DDD, BDD, BRD
* Developers would prepare the Source Code [bit-Bucket, git-hub]
* Software, hardware, and testing occur during the Development Stage
* A Contingency Plan is also developed at this point
* An emergency management document. If the power goes out- what happens to the system? What is the back up? How fast can it be brought back up to speed? Again, documentation and approval (get used to this)

\*A contingency plan is a back-up plan

Testing Phase

* Review and study all available/ provided documents – Use Case, User Stories, FRS, SRS, BDD, TDD, DDD, Project Plan/Charter, Meeting Minutes
* Create Software Test Plan, test Strategy
* Arrange Requirements Walkthrough Meetings, Test Plan Walkthrough Meetings etc. with BA/PM
* Test Design Techniques and Artifacts
* Create scenarios, requirements, Test Case, Test Script
* Software Testing Life Cycle (STLC)
* Bug/Defect Life Cycle (BLC)

\*High-level view is an overview outline

\*Low-level view is a more detailed view of a certain section from the overview outline

\*Test design Technique & Artifact are how you will design your test cases

Delivery & Maintenance Phase

* Operational Level Agreements (OLA) – roles and responsibilities of individuals/tams during delivery
* Production Implementation Plan – Product delivery steps, tech support , hardware/software components, roll back/contingency plan
  + Training users/Knowledge transfer – Introduce system/new features
* New or enhanced system is installed in the production environment/Go-Live
* A “transition” or “cut-over” plan – data conversion
* A ‘Release Notes’ sent to the team and upper management

Scrum Roles

* Product Owner
  + A representative from the Business/Client
  + Creates and prioritize product backlog
  + Manages releases, describes features to the team
* Scrum Master
  + Responsible for team to follow Scrum/Agile values
  + Runs Sprint Planning & Daily Scrum Meetings
  + Ensures team productivity & efficient communication
  + Guards the Scrum team and removes impediments
* Scrum Team
  + Estimate and implement features
  + Sprint Backlog → Shippable products
  + Track work progress everyday – Daily Scrum Meeting (Daily Stand Up)
  + Communicate with product owner/scrum master/leads & alert when there are problems

Scrum Ceremonies

* Sprint Planning Meeting
  + Selection, discussion & estimation of features from Product Backlog for current sprint
  + PO describes the details of features & answers questions
  + Team estimates & commits to the feature
  + Usually scheduled for 2 hours
  + Usually between Tuesday-Thursday
* Daily Stand Up/Scrum Meeting
  + Status of work progress meeting – Same time, same place, everyday for 10-15 minutes, stand-up, no problem solving
  + Each team member answers 3 questions
    - What have I done? – Status/Progress since yesterday/Last meeting
    - What am I going to do? – What’s in queue for today/next meeting
    - What problems do I have? – Anything I need to accomplish my tasks, dependencies on another team/member
  + Issues/concerns recorded by Scrum Master & handled after the meeting
  + Stakeholders are invited to observe but can’t talk – ‘pigs can talk’
* Sprint Review Meeting
  + Demonstration of implemented features/defect fixes
  + ‘Demo’ to Stakeholders/Business – done after each sprint
* Sprint Retrospective Meeting
  + Discussion of goods and bads of the sprint
  + Constructive criticism – people, relationship, process & tools
  + What went well, what didn’t & areas of improvement
  + Actionable suggestions to improve performance
  + Hosted & moderated by Scrum Master
  + Meeting time-based to 1-2 hours

QA Role in Agile

* Continuously study & review all project documentations
* Attend various Project/Scrum Meetings & keep QA Team informed
* Capture complex and negative behaviors on AUT, think beyond the ‘happy path’
* Write TC, create TS, clarify questions/concerns with PO/PM/BA/DEV/SME
* Pair up with developers for reproducing Defects, writing Unit Test cases and for discussing QA Test scenarios
* Perform Negative, Boundary & Smoke/Ad-Hoc Testing formally/informally
* Be available & Flexible in providing immediate support to Devs on retesting Defect fixes & run Regression Test Cases/Scripts/Suites
* Stay focused and informed of new User Stories, Defect Status, Test Deliverables
* Maintain the criteria for DoD and improve quality of AUT
* Maintain all test Artifacts – Test Strategy, Test Plan, RTM, TC, Minutes, etc

\*DoD – Definition of done