Multi-tier Software Development

PRACTICE PLAN

SIA&SDBIS

Practice Plan

- 1. Practice Teams
- 2. Project Themes
- 3. Practice Terms

1. Practice Teams

- Complete Team structure:
  - 3 members: analysts, developers and testers
    - 1 team leader

- Each team project
  - will have a Git account and folder:
    - e.g.: Github.com, Gitlab.com, BitBucket.com

2. Project Themes and modules Proposed Examples

- Software Management Suite
  - I. Software/Service Development Activities Management SAM
  - II. Software Product/Release Management SPM
  - III. Software/Service Quality Assurance Management SQM
  - IV. Team Agile Management
- Other project management or business areas.

I. SAM Modules
Software Development Activities Management

- SAM1: SDLC Management (Software Development Lifecycle Management):
  - Projects, phases, activities, artifacts (including source code modules and documentations), tools, roles, teams, members
- SAM2: Feature/release implementation and evolution:
  - Feature lists, projects/subprojects and features (stories) approached in iterations and releases.
- SAM3: Management of Iterations/Activities:
  - project, iterations, activities (tasks) assigned to team members and project evolution tracking, team, members
- SAM4: Budget/resources Management:
  - time budget, financial budget on projects, iterations, releases, tasks
- SAM5: Documentation Management:
  - Design/UML Docs, API Docs, User Handbooks, Demos, Customer Feature Guides in accord with software releases.

II. SPM Modules
Software Product Release Management

- SPM1: Feature Management
  - products, services, solutions, versions, editions, pricing, deploying-devices
- SPM2: Contract Management
  - clients, contracts, supporting, licences, warranties
- SPM3: Release Configuration Management
  - sites, installations, platforms, versions, comp-dependencies
- SPM4: Product marketing/advertise
  - demos, workshops, presentations, expositions, contacts, trials, events
- SPM5: Customer Acceptance and Support
  - trainings for customer-admins and end-users,
  - customer site deployment process,
  - admin services (remote or on-site).

III. SQM Modules
Software Quality Assurance Management

- SQM1: Build Test Management
  - modules and unit tests, integration tests, e2e tests, acceptance tests
- SQM2: Performance Optimizing
  - benchmarks, datasets workloads, performance metrics, performance tests and results, diagnosis reports, improvement solutions
- SQM3: Customer Feedback Management
  - complaints (downtimes, incidents), improvements, change requests, bugs, assistance activities, warranty tracking
- SQM4: BugFix Management (bug tracking)
  - bugs reporting/recording, description, assignment, implementation, testing;
- SQM5: Quality Audit Procedures
  - audit type, audit criteria and procedures
  - audit investigation, audit subjects/objects, plan, activities, reports

IV. TAM Modules
Team Agile Management

- TAM1: Interviews Management:
  - candidates, interviews, tests, questions;
- TAM2: Scrum/Kanban meetings/retrospectives management:
  - meetings, participants, topics, action items;
- TAM3: Scrum/Kanban Boards:
  - epics, stories, tasks, subtasks, teams, statuses (to dos,in progress, done);
- TAM4: Internships:
  - students, start/end, teams, projects, benefits, schedules;
- TAM5: Career Path management:
  - employees, positions, functional departments, evaluations (quarterly), benefits.

3. Practice Terms

- Each term targets a Sprint about 2 weeks
  - that will have a concrete output to *review* and to be *evaluated*.
- Practice Project stages:
  - Team building, theme setting (including high level requirements).
  - From data requirements to data structures.
  - From data sources to data services.
    - First Evaluation Term
  - Build web data services to access data sources.
    - Second Evaluation Term
  - Build web clients as data service consumers.
  - Finalize web applications.
  - Resuming (third evaluation term):
    - Final Deploying
    - Final Assessment

Practice Flow

- Start Assignment
- Deliver Documentation and Plans
- Deliver source code (modules) on integrated version system
- Deploy components on integration server/platform
- Assessment:
  - check documentation;
  - check source code;
  - run tests on integration server.