DADock Workshop Overview (Proposal)

* Common Technology Edition (Classroom Lecture)
  + Common
    - Ticket Management
    - Source Code Version Management
      * Branch Strategies of Git
    - Communication Platform
  + CI
    - Auto-build
    - Auto-test
      * Unit Test
      * Integration Test
    - Auto-analysis of quality
  + CD
    - Auto-deploy
    - Auto-monitoring
  + DADock Demonstration
    - Actual demonstration showing how the above functions are operated
* Git Edition (Classroom Lecture)
  + Trimmed-down version from both Git trainings above
* GitLab Edition (Classroom Lecture)
  + GitLab
    - History of GitLab
    - Coverage of GitLab
    - GitLab Edition
    - Comparing GitLab with other tools
    - Major users of GitLab
  + Overview of Major Functions and Portal Demonstration
    - Group
    - Project
    - Commit/Merge Request
    - Issues
    - Review
      * Reviewing at MR
      * Reviewing Apps
    - CI/CD Pipeline
      * Overall Image (such as GitLab Server, Runner)
      * Configuring .gitlab-ci.yml
      * Configuring pipeline execution for every pattern
        + Executing pipeline for each commit
        + Executing pipelines regularly
        + Changing pipeline process for each branch
        + Executing pipeline process only when there is an assigned tag
        + Retrieving build results from other PRJs
        + Describing deploy process
        + Building CI/CD pipeline bundled with multiple projects
        + Others
      * Artifact Function
        + Linking with Pages
    - Additional) Pages
    - Additional) Container Registry
    - Additional) Docker Type Runner
    - Additional) LFS
    - Additional) Auto DevOps
  + Case Study Introduction
    - Large-scale GitLab Repository – from GitLab.com
* Gradle Edition (Classroom Lecture)
  + Gradle
    - Historical Background (make → ant→ maven→ gradle)
    - Task Runner(Build tool)
    - Task Runner Functions & Roles
    - Additional) Introducing representative Task Runners for each language/FW
    - Gradle Features
    - Additional) Resolving dependencies
    - Comparing other Task Runners
    - Additional) Gradle Wrapper
  + QuickStart (Easy Installation Procedures)
  + Plugin
  + Source Set
  + Task
  + Gradle Script
    - Definition of various files (such as .gradle, .properties) and execution sequence
    - Configuration overview of various files
  + Basic Notations
    - How to define task
    - How to define dependency relationship
    - How to define task rule
    - Controlling task
    - Executing task conditions
    - Attaching task sequence
    - Finalizing task
  + Plugin
  + Additional) Introducing alternate plugins
    - java
    - eclipse
    - war
  + Illustrating described case studies for each pattern
    - Building independent Java application
    - Building Web application
    - Building/executing JUnit test
    - Creating JavaDoc
    - Executing SonarQube source code analysis
    - Configuring multiple projects
    - Storing jar in the Artifact management tool
    - Operating file
    - Additional) Gradle over Proxy
    - Application) Spring at Gradle
    - Application) Java on Docker at Jib
  + Links to Case Studies
    - Large-scale case study which used this tool (Gradle) was retracted and linked from GitHub
    - ElasticSearch, etc.
* JUnit Edition (Classroom Lecture)
  + Introduction to Test Automation
    - Test Automation
      * Different phases in Test Automation (such as generate automation items, generate automation test, execute automation test, compare automation test results)
      * Executing test automation and comparing results will be handled in this workshop
    - Why do we automate the testing?
    - Automating Unit Test
  + JUnit
    - Overview
    - xUnit
    - XML of JUnit
    - JUnit 5
  + Important points in Test Automation
    - Principles in Test Code
    - Writing readable test codes
  + QuickStart (Easy Installation Procedures)
  + Basic Functions
    - Assertion
    - Structuring the Test
    - Parameter Test
    - Categorizing Test
      * Slow Test Problem
    - Assumption
    - Fixture
    - Rule (Introducing Various Rules)
      * Temporary file operation, Set of default items
  + Reference) Database Test
  + Test Double
    - Testability
    - Test Double
      * Mock, Stub, Spy
    - Mockit Introduction
  + Code Coverage
    - Coverage
    - Coverage Goals
    - Should the coverage be set to 100%?
    - Points to Note
  + Test-driven Development
  + Additional
    - Additional) When using from Maven
    - Additional) When using from Gradle
    - Additional) Cool functions of Junit 5
  + Links to Case Studies
    - Large-scale case study which used this tool (JUnit) was retracted and linked from GitHub
* Selenium Edition (Classroom Lecture)
  + Selenium
  + Which is good among types of Selenium?
    - Selenium IDE
    - Selenium Web Driver
    - Selenium Remote Control
    - Selenium Grid
  + Selenium Architecture
    - Client/Server/Grid
  + QuickStart (Easy Installation Procedures)
  + Illustrating described case studies for each pattern (Selenium WebDriver)
    - Generating and deleting browser
    - Setting specific driver
    - Retrieving/Operating Components
    - Retrieving/Operating Component Information
    - Retrieving/Operating Browser Information
    - [Important] Wait Processing
    - Operating pop-ups/windows/frames
    - Executing various operations (such as click, input)
    - Uploading/downloading files
    - Basic authentication dialogue
    - Obtaining screenshots
  + Setting up Various Drivers
    - Such as Chrome/Firefox/IE/Edge
  + Application) Page Object Pattern
    - Page Object
    - Page Object Merit
  + Application) Integrating with CI Tools
    - Application) Integrating with JUnit
    - Application) Integrating with Gradle
    - Application) Integrating JUnit with GitLab CI
  + Building Selenium Servers
    - Easy-build using Docker
    - How to build Selenium Grid
  + Introducing Various Related Tools
    - Introducing convenient extension wrappers (like Geb)
    - Introducing tools for Mobile (Appium)
  + Links to Case Studies
    - Large-scale case study which used this tool (Selenium) was retracted and linked from GitHub
* Ansible Edition (Classroom Lecture)
  + Ansible
    - Background (like Infrastructure as code)
    - Overview
    - Features
    - Comparing with other tools
  + Architecture of Ansible
    - System Configuration of Ansible
    - Module
    - Playbook
    - Inventory
    - Plugin
  + QuickStart (Easy Installation Procedures)
  + Basic Knowledge
    - Inventory
    - Playbook
  + Illustrating described case studies for each pattern
  + Bootstrapping: Installing to Starting the OS
    - Cloud API (example of OpenStack)
    - Linking with Docker
  + Configuration Management: OS Basic Settings to Mid Installation
    - Linux
    - Windows
  + Orchestration: Deploying Applications to Setting Links
    - Java Web
    - C# Web
  + Application)Practicing Infrastructure as Code
    - Applying Ansible
      * Playbook Best Practice
      * Performance Tuning
      * Troubleshooting
      * Encrypting
    - Ansible Tower(AWX)
    - Integrating with CI
  + Links to Case Studies
    - Large-scale case study which used this tool (Selenium) was retracted and linked from GitHub
* SonarQube Edition (Classroom Lecture)
  + SonarQube
    - Background
    - Overview
    - Features
    - Supported language and available coverage in Free version
    - Comparing with other tools
  + SonarQube Architecture
    - Overall Structure Image
    - SonarScanner
    - Sample structure image (Java)
  + Function Overview
    - How to read various metrics
    - Time series metrics
    - Plugin
  + How to Use each Pattern
    - Adding plugins
    - Creating quality profiles (Quality Gate)
    - Marking issues with no problem
    - Overwriting results of different kinds of test (Unit Test, Integration Test)
    - Analyzing quality of C# Project
  + Links to Case Studies
    - Large-scale case study which used this tool (SonarQube) was retracted and linked from GitHub
* Mattermost Edition (Classroom Lecture)
  + Mattermost
  + Function Overview
    - Channel
    - Mention
    - Flag, Pinning contents
    - Emoticons
    - Permalink
    - Integrated Functions (Slash command, Web hook)
    - Other detailed functions
  + Best Practice
    - Viewpoints in encouraging information sharing
    - Sample channel allocations
    - How to jump-start/invigorate Best Practice
  + Application Edition
    - Integrating with GitLab
    - Trying out different integrations