



**Hackathon model POC – evaluation parameters, expectations, expected questions, acceptance criteria**

## TABLE OF CONTENTS

1. INTRODUCTION
2. EVALUATION PARAMETERS OF POC
3. EXPECTATIONS AND EXPECTED QUESTIONS
4. HIGH LEVEL ACCEPTANCE CRITERIA



## 1. INTRODUCTION

### OVERVIEW

The e-Pragati Authority has raised an RFP for the purchase of low-code or no-code platform, henceforth will be known as e-Pragati Core Platform (ePCP).

This document will provide high level evaluation criterion, POC acceptance criterion, expectations and questions you could expect from e-Pragati evaluators. A [more granular level details of the information provided in this document](#) along with detailed problem statement for the proof of concept (POC) will be provided on the day of Hackathon.

It's expected that each team will build the services / applications and demonstrated to e-Pragati technical evaluation team.



## 2. EVALUATION OF THE POC DESIGN

### PRE DEMO EVALUATION

- Problem Definition & Solution Engineering
- Implementation team composition
- Implementation team co-ordination and execution practices followed for implementing the solution
- Platform capabilities demonstration
  - Ease of design
  - Ease of development
  - Ease of deployment
  - Ease of change/enhancement

### DURING DEMO EVALUATION

- Presentation
  - Overview of topics/presentation
  - Identified critical concepts (best practices)
  - Explanation of technical issues
  - Handling questions
  - Summary, conclusions & critique
  - Resource & Time management
  - Visual Aids
  - Overall impression



## 2. EVALUATION OF THE POC DESIGN

### DURING DEMO EVALUATION

- Demonstration
  - Usability
  - Sustainability and maintainability
  - Governance
  - Accessibility
  - Testability
  - Portability
  - Supportability
  - Analysability (easy to understand at code level)
  - Configurability
  - Evaluability (futuristic or future proof)
  - Interoperability

### POST DEMO EVALUATION

- Innovation, Re-engineering and Process Improvement
  - Number of proposed changes / amendments in the system
  - Utility and improvements resulted as an effect of each change
  - Adherence to global BPR practices
  - Reducing TAT and raising the bar
- Completeness of solution and compliance
  - Completion of mandatory flow or optimal user journey
  - End to End Solution for the proposed problem (demonstrate a high solution engineering capability)
  - Security and user authentication
- Adherence to the proposed Solution

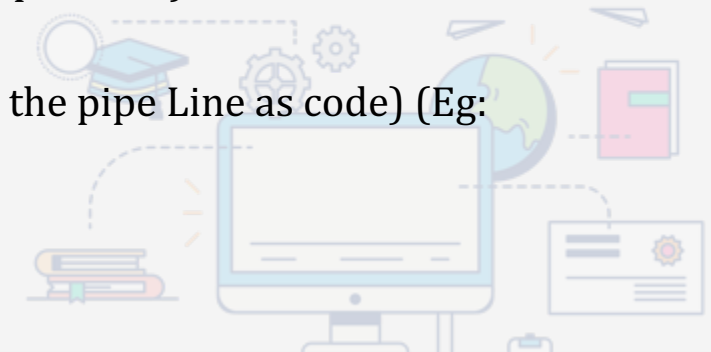
## 3. EXPECTATIONS & EXPECTED QUESTIONS TO HACKATHON TEAM

### Day 1

- Architecture Review
- List of proposed technology stack (along with the mobile stack for responsive apps)
- Proposed Solution / Approach
- Best Practices from the technology stand point

### Day 2

- Show how the design and code is adhering to the proposed architecture
- Build reusable components (mostly from the front end)
- Will the API Repository be maintained ?
- How easy is code maintenance (configuration driven enhancements, reusable components )
- Clear vision of the open source communities to the open sources used
- DevOps Maturity Model (this has to be configurable that can be achieved defining the pipe Line as code) (Eg: Jenkins with plugins like Blue Ocean )



## 3. EXPECTATIONS & EXPECTED QUESTIONS TO HACKATHON TEAM

### Day 2 (contd.)

- How the licensing and the vulnerabilities report is generated (this is very important from the open source licensing perspective)
- How security of the sensitive data will be taken care ?
- How mobile data that is saved on the device is taken care (Eg: use of encrypted databases like Realm) ?
- Is the OWASP top 10 vulnerabilities considered while coding? (for web and mobile)
- How much caching (content static) and In Memory is used ?
- Envisioning if any custom frameworks can be build for the applications (at the platform level) ?
- Comparison of the tech stack if any near component is available
- How logs are centralized in the distributed systems?
- Is the Inclination of the architecture is Stateful Clustering vs Stateless (Stateless may have advantages from the perspective of the Infra utilization and costing) ?
- How the legacy data is converted to the current Pattern that is designed ?



## 3. EXPECTATIONS & EXPECTED QUESTIONS TO HACKATHON TEAM

### Day 2 (contd.)

- Submit report of the NFRs factors considered and adhered
- The Application has to scale (services)
- Page loading time has to be less than 3 secs. At the peak Load the load times can go up to 5 Secs. (no services has to go into no responsive state)
- With minimum configuration, any service can talk to any service
- How the analytics are generated ?
- Are the dashboards configurable ?





## 4. HIGH LEVEL ACCEPTANCE CRITERIA

- Ensure that the development of POC adheres to low code or no code principles (no building from scratch)
- Ensure that POC highlights the key features of the ePCP (e-Pragati Core Platform)
- Ensure that developed application supports all the technical parameters mentioned in RFP.
- Ensure that the system works properly in the business environment (the application has to be deployable and in running state)
- Ensure that the design meets the business requirements.
- Test the deployment process for business logic and security



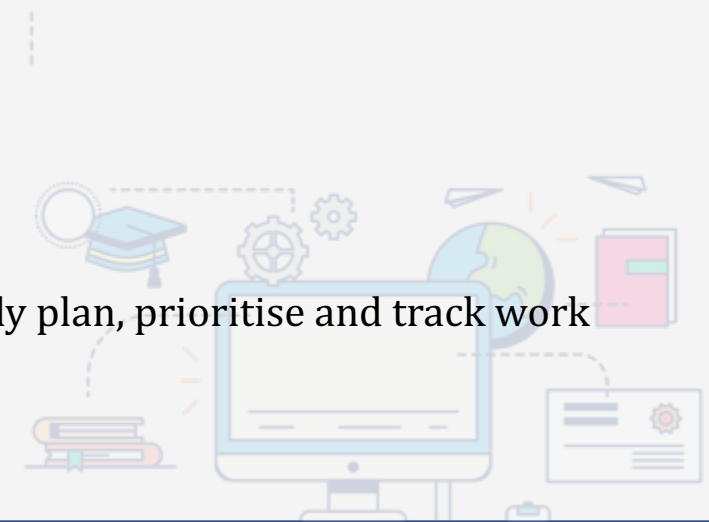
## 4. HIGH LEVEL ACCEPTANCE CRITERIA

- "fully functional" version of the application module built using the platform instance that's inline with e-Pragati architectural vision for core platform
  - e-Pragati application build - deployed artefacts
  - auto-generated data models
  - reusable components
  - use case data flows, integrations & interactions
- Able to see below created using the user story brief and other information provided
  - Solution (functional & technical) Architecture Diagram
  - DFD
  - Maximum utilisation of the available datasets
  - UX approach
    - User Personas
    - Mock Designs
- Able to see code once, run anywhere with minimal configuration changes



## 4. HIGH LEVEL ACCEPTANCE CRITERIA

- Provide insight into DevOps stack (Tools and the Tool gates used as part of the pipe line)
- Testing approach
  - Manual
  - Automation
  - Performance
- Documentation
  - User level (User Guide / admin guide)
  - Release notes (per release)
  - Training / Capacity building
- Scale agile development practices
- SaaS-based instance for the e-Pragati - provide a hub for teams to collaboratively plan, prioritise and track work



# THANK YOU

