

Transparency-One Technical test - Pristine version for candidates

Contexte

Le candidat se place dans la situation d'une code review dans le cadre d'une codebase à l'échelle entreprise avec des problématiques de maintenabilité et scalabilité long terme. La codebase correspond à un projet web.

Extrait de code à commenter

```
using System;
using System.Data.Entity;

namespace Nasa.Web.Controllers.Api
{
    public class SpaceShuttle
    {
        public long? Id { get; set; }
        public string color { get; set; }
        public AssemblyLine type { get; set; }
        public long mileage { get; set; }

        private object thrustEngine;
        private object turnEngine;
        private int oilLevel;

        DbContext dbContext = new DbContext("myConnectionString");

        public SpaceShuttle(long? id, string color, AssemblyLine type)
        {
            this.Id = id;
            this.color = color;
            this.type = type;
        }

        public void Save()
        {
            if (this.Id == null)
            {
                this.dbContext.SpaceShuttles.Add(new SpaceShuttleEntity
(this));
            }
            else
            {
                this.dbContext.SpaceShuttles.Update(this.Id, this.
color, this.type);
            }
        }
    }
}
```

```

    public async SpaceShuttle Get(long id)
    {
        return this.dbContext.SpaceShuttles.Where(s => s.Id == id).
Select(s => new SpaceShuttle(s.Id, s.Color, s.Type, s.mileage)).
FirstOrDefault();
    }

    public string GetModel()
    {
        return "This shuttle is a " + this.type;
    }

    public SpaceShuttle GetShuttleAndUpdateKms()
    {
        SpaceShuttle shuttle = this.Get(this.id);
        shuttle.mileage = mileage++;
        this.Save();
        return shuttle;
    }

    public bool HasSolarPanel()
    {
        switch (this.type)
        {
            case AssemblyLine.Enterprise:
                return true;
            case AssemblyLine.Columbia1:
                return true;
            case AssemblyLine.Columbia2:
                return true;
            case AssemblyLine.Challenger:
                return true;
            case AssemblyLine.Discovery:
                return false;
            case AssemblyLine.Atlantis:
                return true;
            case AssemblyLine.Challenger:
                return true;
            default:
                return false;
        }
    }

    public bool Start()
    {
        try
        {
            this.thrustEngine.Start();
            var startOk = this.turnEngine.Trigger();
            if (this.oilLevel < 20)

```

```

        {
            throw new Exception("Insufficient oil");
        }
        return true;
    }
    catch (Exception)
    {
        throw new Exception("Engine failure at startup");
    }
}

public static float GetNumberOfPassengers(string type)
{
    if (type == AssemblyLine.Enterprise)
    {
        return 4;
    }
    if (type == AssemblyLine.Challenger)
    {
        return 4;
    }
    if (type == AssemblyLine.Atlantis)
    {
        return 5;
    }
    return 0;
}
}

namespace Nasa.Web
{
    public enum AssemblyLine
    {
        Enterprise,
        Columbial,
        Columbia2,
        Challenger,
        Discovery,
        Atlantis,
        Challenger
    }
}

```

Consignes

Le candidat dispose de la soirée pour effectuer le commentaire de code et envoyer sa réponse par retour d'email.

De la même manière que le test est flexible sur le temps accordé, la forme du retour est également laissée au choix du candidat.

Formes potentielles acceptées :

- Retour de l'extrait de code annoté de commentaires
- Commentaire texte développé dans un fichier séparé
- Implémentation d'un refactoring du code
- Diagrammes UML représentant la forme refactorée du code
- Mix de l'ensemble des formes ci-dessus

L'extrait de code à commenter présente un ensemble de problématiques allant de soucis d'implémentation de surface et d'utilisation du langage C# à des soucis plus profond de conception, architecture, maintenabilité et résilience à des besoins d'évolution future.

Il est primordial que le candidat effectue une analyse en profondeur de l'extrait de code et justifie ses retours en détail. La capacité à identifier les problématiques, les expliquer et proposer des solutions et alternatives rendant le code plus robuste sera au coeur des éléments utilisés pour analyser la qualité du retour effectué.