CitiusTech

SmartCode Initiative Orientation

June 2015

Agenda

- **SmartCode Background and Overview**
- SmartCode Initiative Modules
  - Design Principles
  - Design Patterns
  - Coding Guidelines
  - QMS Enhancement
  - Coding Tools
- SmartCode Audit
- Help from DLs & BLLs in Driving SmartCode Initiative

SmartCode: Background

Key Business Realities:

- We are a **rapidly growing** company….our headcount has increased almost 3 times in the last 2 years
- Healthcare is a **life-critical domain** - quality of our deliverables has a direct impact on human life
- Most of CitiusTech's clients are **best-in-class healthcare ISVs** with strong internal teams and high expectations

Implications for CitiusTech:

- It is very important for us to define and maintain strong internal processes to assure the quality of CT deliverables – esp. the quality of the software code that is produced by CT engineers
- Our ability to implement strong processes and consistently produce high quality software code across different languages/technologies will be the key success factor for us as we evolve as a company

Enhancing & maintaining software quality is the PRIMARY KRA for all of us

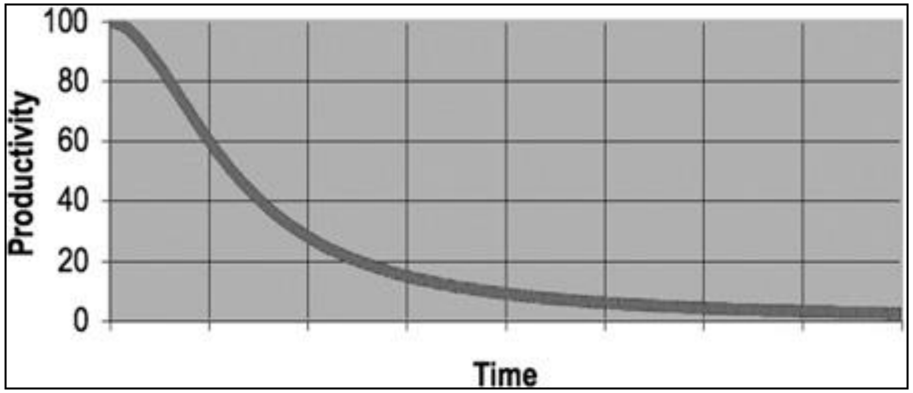
SmartCode: Overview

- The SmartCode is an organizational initiative to enhance and maintain the quality of software code produced by CitiusTech
- The initiative has involved several key aspects - e.g. defining coding guidelines, identifying automated code review tools, diligently implementing review processes, conducting internal training and maintaining continuous awareness in our teams
- To meet this objective, we formed a SmartCode CTP in which we closely worked on with the Training teams and created comprehensive training material (e.g. Design Principles, Coding Guidelines). These are now part of the UniverCT CTCourses
- We are working with the training team to roll out SmartCode trainings along with the SmartCode Audit process to the pilot teams first and then to wider audience
- Objective for today's session is to share the highlights of the SmartCode and request for your help as we roll out the SmartCode as a cross-functional initiative and systematically review and track our progress

The Objective of the SmartCode Initiative is to create a foundation that will enable all our project teams to produce best in class code for our clients

SmartCode: Why?

Poor Code impact: As the mess builds, the productivity of the team continues to decrease, asymptotically approaching zero. Reference: Clean Code by Robert C Martin



The Boy Scout Rule: ***Leave the campground cleaner than you found it.***
It's not enough to write the code well. The code has to be kept clean over time. We've all seen code rot and degrade as time passes. So we must take an active role in preventing this degradation.

SmartCode Rule: Check-in our code a little cleaner than when we checked it out
The cleanup doesn't have to be something big. Change one variable name for the better, break up one function that's a little too large, eliminate some duplication, clean up one composite if statement.

CitiusTech

Agenda

- SmartCode Background and Overview
- **SmartCode Initiative Modules**
  - Design Principles
  - Design Patterns
  - Coding Guidelines
  - QMS Enhancement
  - Coding Tools
- SmartCode Audit
- Help from DLs & BLLs in Driving SmartCode Initiative

SmartCode: Training Modules

Module	Sessions	Hours	Objective
Design Principles (Common)	1	2	• To learn design principles (incl. SOLID principles) • To understand the importance of good design.
Design Pattern (Common)	3	6	• To learn 9 commonly used design patterns and their applicability. • Understanding the benefits to use known solutions that are tried and tested.
Coding Guideline (Tech specific)	1	2	• To learn Standard coding guidelines and best practices which will add quality, readability to software and make the maintenance phase easy.
Tools (Tech specific)	0.5	1	• To learn how usage of code review tools help us to automate the review process.
QMS (Common)	0.5	1	• To learn how to Configuring Coding & Review guidelines in QMS. • To learn how to Conduct Work Reviews on QMS.
Total	6	12	

Agenda

- SmartCode Background and Overview
- **SmartCode Initiative Modules**
    - **Design Principles**
    - Design Patterns
    - Coding Guidelines
    - QMS Enhancement
    - Coding Tools
- SmartCode Audit
- Help from DLs & BLLs in Driving SmartCode Initiative

SmartCode: Design Principles (1/2)

- Due to the dynamic nature of current projects its important to understand:
  - What is object oriented design?
  - What are its benefits ?
  - What are it's costs?
- Uncle Bob (Robert C. Martin) has identified SOLID principles of object oriented software design. These principles govern the structure and interdependencies between classes in large object oriented systems
- These principles when applied together intend to make it more likely that a programmer will create a system that is easy to maintain and extend over time
- The principles of SOLID are guidelines that can be applied while working on software to remove code smells by causing the programmer to refactor the software's source code until it is both legible and extensible
- It is typically used with test-driven development, and is part of an overall strategy of agile and adaptive programming

SmartCode: Design Principles (2/2)

- **Scope**
  - Cover 5 Design principles
  - Emphasize the need and importance of each principle supported by video

Key five principles of class design covered		
SRP	The Single Responsibility Principle	A Class should have one and only one reason to change
OCP	The Open Closed Principle	You should be able to extend classes behaviour, without modifying it
LSP	The Liskov Substitution Principle	Derived classes must be substitutable for their base classes
ISP	The Interface Segregation Principle	Make fine grained interfaces that are client specific
DIP	The Dependency Inversion Principle	Depend on abstractions, not on concretions

Agenda

- SmartCode Background and Overview
- **SmartCode Initiative Modules**
  - Design Principles
  - **Design Patterns**
  - Coding Guidelines
  - QMS Enhancement
  - Coding Tools
- SmartCode Audit
- Help from DLs & BLLs in Driving SmartCode Initiative

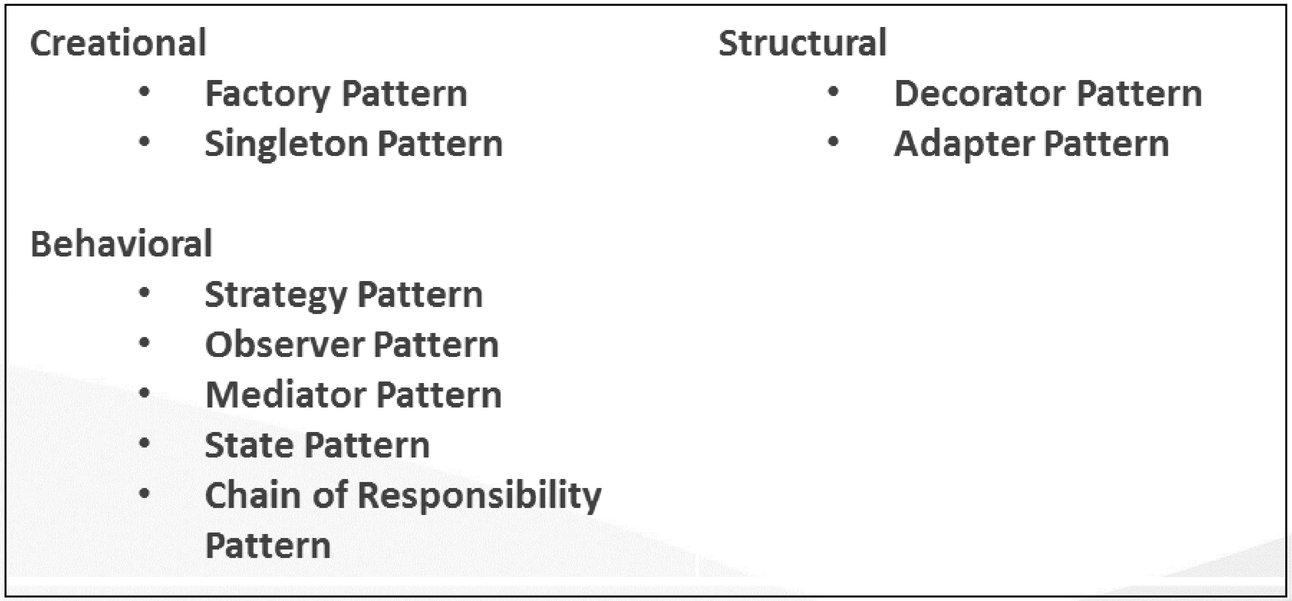
SmartCode: Design Patterns (1/2)

- **What are Design Patterns ?**
  - Design patterns are optimized, reusable solutions to the programming problems that we encounter every day
  - A design pattern is not a class or a library that we can simply plug into our system; it's much more than that

- **Training emphasizes on "Why should we use them ?"**
  - Design patterns are well proven existing solutions to programming problems. So developers have not to re-invent the wheel
  - Design pattern add quality to you code and help in reducing the cost involved in maintenance phase

SmartCode: Design Patterns (2/2)

- **Scope**
  - Design patterns were originally grouped into the categories: Creational patterns, Structural patterns, and Behavioral patterns, and described using the concepts of delegation, aggregation, and consultation. We are going to cover nine design patterns
  - In this session trainer would be discussing each design pattern supported by videos
  - Trainer would try to make the session interactive as much as possible



Creational
- Factory Pattern
- Singleton Pattern

Structural
- Decorator Pattern
- Adapter Pattern

Behavioral
- Strategy Pattern
- Observer Pattern
- Mediator Pattern
- State Pattern
- Chain of Responsibility Pattern

SmartCode: Evaluation test for Design pattern

- At the end of the Design pattern session participants would be given 10 real world scenarios .They need to identify appropriate design pattern for each scenario
- Below is the sample snapshot of business scenario:

Application has to be made for Air Traffic control. An airport control tower looks after who can take off and land – all communications are done from the airplane to control tower, rather than having plane-to-plane communication. This idea of a central controller is one of the key aspects of Air traffic control mechanism. Identify the design pattern to be used for this problem statement.

Agenda

- SmartCode Background and Overview
- **SmartCode Initiative Modules**
    - Design Principles
    - Design Patterns
    - **Coding Guidelines**
    - QMS Enhancement
    - Coding Tools
- SmartCode Audit
- Help from DLs & BLLs in Driving SmartCode Initiative

SmartCode: Coding Guidelines (1/2)

- Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language
- Reducing the cost of software maintenance is the most often cited reason for following coding conventions
- In this session we will emphasize why Coding conventions are important to programmers and also focus the below reasons:
  - 40 – 80 % of the lifetime cost of a piece of software goes to maintenance
  - Hardly any software is maintained for its whole life by the original author
  - Code conventions improve the readability of the software, allowing engineers to understand new code more quickly and thoroughly
  - If you ship your source code as a product, you need to make sure it is as well packaged and clean as any other product you create

SmartCode: Coding Guidelines (2/2)

- Coding Guidelines for various technologies are stored in interCT under Competency > TLG > Software Engineering 2
- Also, to effectively implement the Code review process as part of the SmartCode initiative, these coding guidelines are integrated into QMS is updated
- It provides with two key capabilities:
  - Configure Coding guidelines in QMS as a checklist (covered later)
  - Use coding Checklist during the code review and upload process (covered later)

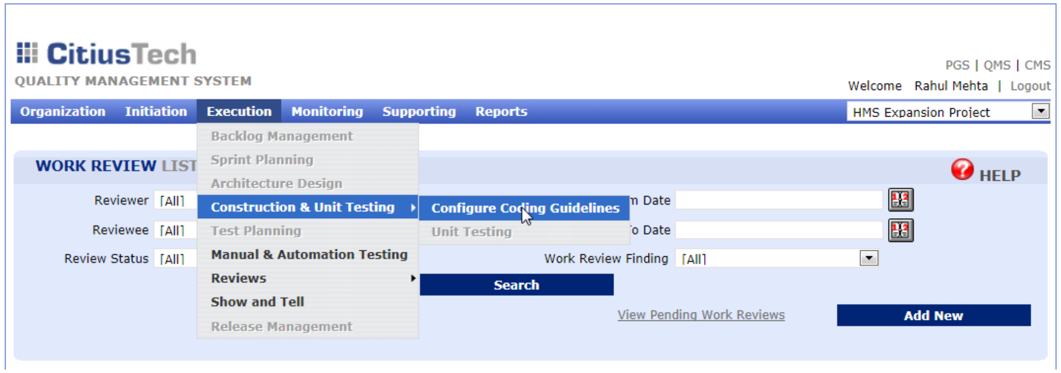
Agenda

- SmartCode Background and Overview
- **SmartCode Initiative Modules**
    - Design Principles
    - Design Patterns
    - Coding Guidelines
    - **QMS Enhancement**
    - Coding Tools
- SmartCode Audit
- Help from DLs & BLLs in Driving SmartCode Initiative

SmartCode: QMS – Configure Code Review Guidelines

- Coding Guidelines need to be configured for EACH QMS Projects at the beginning of the project for all applicable technologies in QMS

QMS →Execution →Construction & Unit Testing →Configure Coding Guidelines → Add Coding Guidelines



- Default organization level templates of **.NET, Java, JavaScript & SQL** have been uploaded in QMS. This can be customized to the project needs. Contact Quality Team for **Other technologies & QA projects** – to create additional checklist

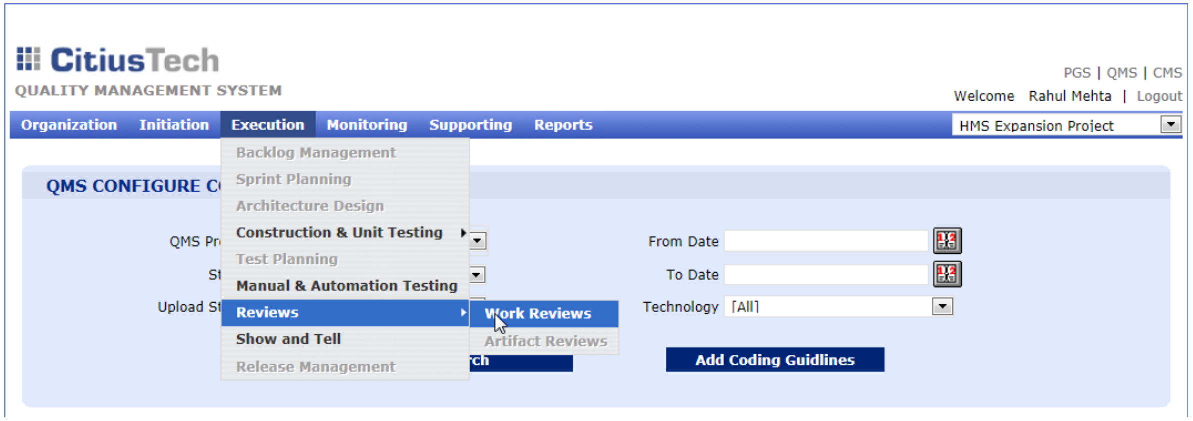
- Video for configuring coding guidelines is available at **InterCT→ UniverCT→CT Courses →Corporate →QMS →Lab**
  http://interct/sites/univer%20ct/OnlineCourses/SitePages/CourseDetails.aspx?CT=Corporate&CS=QMS&CTID=5&CSID=40

  Note: PGS/QMS application works well with Mozilla Firefox. Some functionality may not work in other browsers like IE, Chrome, etc.

SmartCode: QMS – Conducting Code Reviews

- Once Coding Guidelines are configured, they need to be used for conducting work reviews in QMS

QMS→Execution →Reviews→ Work Reviews



- Logging of reviewers comments for the review conducted & tracking to closure of comments is built in the workflow
- Video on conducting work reviews using coding guidelines is available on :

  InterCT→ UniverCT→CT Courses →Corporate →QMS →Lab

  http://interct/sites/univer%20ct/OnlineCourses/SitePages/CourseDetails.aspx?CT=Corporate&CS=QMS&CTID=5&CSID=40

  Note: PGS/QMS application works well with Mozilla Firefox. Some functionality may not work in other browsers like IE, Chrome, etc.

Agenda

- SmartCode Background and Overview
- **SmartCode Initiative Modules**
  - Design Principles
  - Design Patterns
  - Coding Guidelines
  - QMS Enhancement
  - **Coding Tools**
- SmartCode Audit
- Help from DLs & BLLs in Driving SmartCode Initiative

SmartCode: Tools

- We can write high quality code using Code Refactoring and Code Review Tools
    - Tools help us to follow standard coding conventions which increases code readability and in turn helps in maintaining the code
    - Tools do “Code review”. .i.e. developer can do Self review using these tools
    - Tools can do static analysis
    - Code coverage tools help us to test branch coverage
- As part of SmartCode, we will standardize use of popular tools (based on technology) which can add quality to our code or help in mainlining the code

Java	.Net	JavaScript
eCobertura	Resharper	JSLint
FindBugs		
CheckStyle		
PMD		

- These tools with baseline configuration will be stored in a centralized controlled location for everyones use. IT team will help everyone install the tool

Agenda

- SmartCode Background and Overview
- SmartCode Initiative Modules
    - Design Principles
    - Design Patterns
    - Coding Guidelines
    - QMS Enhancement
    - Coding Tools
- **SmartCode Audit**
- Help from DLs & BLLs in Driving SmartCode Initiative

SmartCode: Audit Process and Checklist

- The objective of this audit post the SmartCode sessions is to:
  - Ensure the teams have understood SmartCode principles
  - Demonstrate that they are actively applying the learning from these sessions and
  - Ensure the use of available tools
- SmartCode Audit process is divided to cover five areas:
  - Assess the understanding and effective Use of Design Principles / Patterns
  - Demonstrate implementation of Design Reviews
  - Use of Coding Guidelines for various technologies
  - Use of appropriate code review tools
  - Effective use of QMS
- Each project will receive a Pass/Fail/NA score in these 5 areas based on adherence to SmartCode guidelines and also any improvement suggestions from the Auditors.
- SmartCode Audit template is stored at **InterCT→ Competency → TLG → Software Engineering 2**

Agenda

- SmartCode Background and Overview
- SmartCode Initiative Modules
  - Design Principles
  - Design Patterns
  - Coding Guidelines
  - QMS Enhancement
  - Coding Tools
- SmartCode Audit
- **Help from DLs & BLLs in Driving SmartCode Initiative**

SmartCode: Help Needed

- Please treat this initiative with utmost importance
- SmartCode trainings are **mandatory** for the project teams
  - **100% Attendance** is must
  - If the participants are not able attend the training due to any reason, they must inform the training team and take your prior approval
- These are in-person class room trainings and all participants are required to be **physically present** at training locations irrespective of employees base location. Gotomeeting would not be used
- After training DLs/BLLs would **Audit** their own project in order to assess the efficacy of Smart code Initiative on their respective projects

THANK YOU