
システム開発演習

【オリエンテーション資料】

1. システム開発演習の目的

ねらい

業務では、研修で学んだそれぞれの内容を状況に応じて組み合わせ、対応していく力（**応用力**）が求められます。

「システム開発演習」を通じて、これまで習得したスキルを組み合わせ、プログラミング開発における「**インプット力**」「**アウトプット力**」のコアスキルを高めます。

到達目標

- **開発標準に則ったシステム開発の設計／構築を行うために必要となる各種スキルをしっかりと定着させる**
- システム開発プロジェクトの実践を通じて以下のコアスキルを高める
 - ・顧客へのヒアリング、必要に応じた顧客との折衝
 - ・プロジェクト活動の実施
 - ・各工程に対応した設計書等ドキュメントの作成
 - ・JSP／Servletを使用したWebアプリケーション開発、テストの実施
 - ・DBMS（PostgreSQL）を使用したデータ管理
- 個人でのプロジェクトを進めることで、**組織活動との違い、自主性、積極性、責任感、挑戦意欲**の必要性を体感で理解する
- プロジェクト内で例外的（仕様追加／変更）の場面を作りだし、**応用力**の必要性を体感で理解する
- 顧客／上長役からのフィードバックを行い、**客観的**に物事がどのように評価されるかを感じ、**他者目線**の重要性を理解する

2. スケジュール

全体スケジュール

月日	カリキュラム	月日	カリキュラム
5月12日(金)	設備構築	6月4日(日)	
5月13日(土)		6月5日(月)	コーディング／単体試験
5月14日(日)		6月6日(火)	コーディング／単体試験
5月15日(月)	設備構築	6月7日(水)	結合試験
5月16日(火)	設備構築	6月8日(木)	総合試験
5月17日(水)	設備構築	6月9日(金)	総合試験
5月18日(木)	設備構築（修了テスト）	6月10日(土)	
5月19日(金)	開発標準との違い／開発業務理解	6月11日(日)	
5月20日(土)		6月12日(月)	総合試験
5月21日(日)		6月13日(火)	納品作業／成果発表準備
5月22日(月)	UML基礎	6月14日(水)	成果発表会
5月23日(火)	SVN／キックオフ／プロジェクト計画	6月15日(木)	
5月24日(水)	システム全体計画(SP)／外部設計(ED)	6月16日(金)	
5月25日(木)	システム全体計画(SP)／外部設計(ED)	6月17日(土)	
5月26日(金)	内部・PG設計(ID・PD)	6月18日(日)	
5月27日(土)		6月19日(月)	
5月28日(日)		6月20日(火)	
5月29日(月)	内部・PG設計(ID・PD)	6月21日(水)	
5月30日(火)	内部・PG設計(ID・PD)	6月22日(木)	
5月31日(水)	内部・PG設計(ID・PD)	6月23日(金)	
6月1日(木)	コーディング／単体試験	6月24日(土)	
6月2日(金)	コーディング／単体試験	6月25日(日)	
6月3日(土)			

※カレンダーはグループ演習時のものです。

3.1 システム開発演習における顧客情報

1. 会社概要

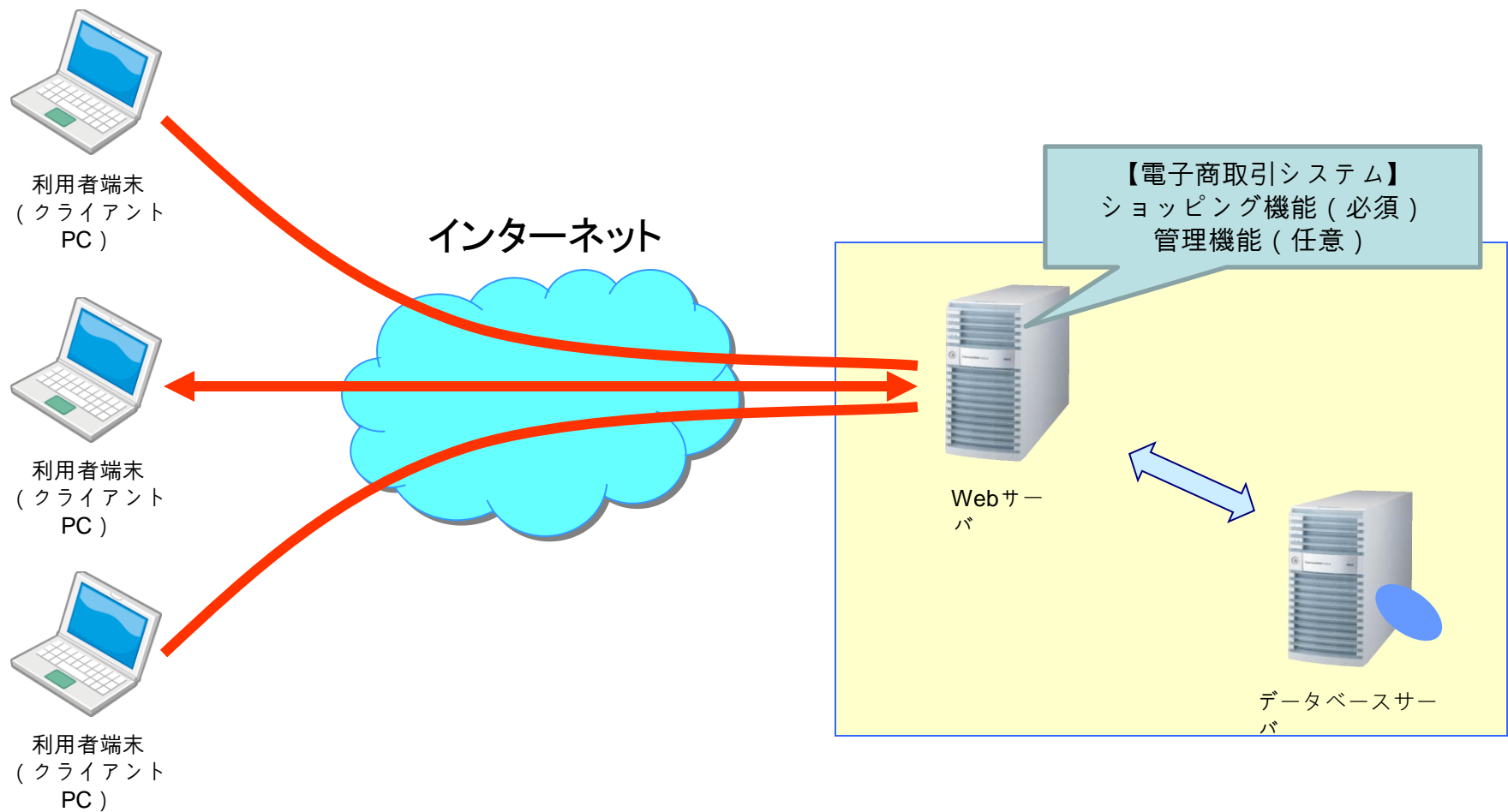
- 会社名 . . . XYZ株式会社
- 業種 . . . 小売業
- 事業内容 . . . 帽子および鞆の販売
. . . etc

2. システム化の背景

現在、店舗販売を中心として営業展開しているが、今後売り上げ拡大のため、オンラインショッピングを利用して全国展開をしたいと考えている。

その為、自社の商品を取り扱うECサイトを作成して販売形態を設けることにした。

3.2 電子商取引システムの構成図概要



3.3 レビュー対象の資料 OUTPUT (納品物)

INPUT (顧客提供物)

1.ドキュメント関連

1.1 システム全体計画

- ・提案依頼書 (RFP)
- ・コーディング規約
- ・開発環境構築手順書
- ・要件定義書
- ・業務フロー

1.2 外部設計

- ・システム化対象業務一覧
- ・画面一覧
- ・画面モックアップ (HTML)
- ・テスト仕様書
- ・テーブル定義書
- ・ER図

2.プログラム関連

2.1 サンプルソース (任意)

⇒サンプルソースは、進捗に応じて確認します。
(模範解答になるため)

OUTPUT (納品物)

1.ドキュメント関連

1.1 システム全体計画

- ・プロジェクト計画書
- ・WBS
- ・目標設定シート

1.2 外部設計

- ・機能一覧 (穴埋め)
- ・画面遷移図 (穴埋め)
- ・イベント定義書 (穴埋め)

1.3 内部/プログラミング設計

- ・クラス図
- ・クラス仕様書

1.4 試験

- ・単体試験計画書兼結果書
- ・結合試験計画書兼結果書
- ・総合試験計画書兼結果書

2.プログラム関連

- ・ソースファイル
ショッピング機能：必須
管理機能：任意

3.品質保証、議事録関連

- ・業務日報
- ・バグ管理簿
- ・議事録

参考：レビューとは

レビューとは(1/2)

■ レビューとは

成果物の品質を確認する手段の一つ。設計書やソースコードについて、記述レベルや記述内容の抜け漏れ、誤りが無いかを関係者で確認するもの

■ レビューの目的

成果物を関係者で客観的に確認することにより、不具合が無いかを確認し、成果物の品質向上を図ること

■ レビューの種別

レビュー種別		内容	レビュー対象
1	内部レビュー (レビュー形式：ウォークスルー) ※ウォークスルー：レビューを希望する作成者が数人のレビューアを招集、成果物の内容を順に説明する形式	・ 成果物の品質向上を目的として、自グループ内で上司レビュー前に実施する ・ レビュー形式は、成果物の作成者が主体となるウォークスルーレビューとする	・ 外部設計書 ・ 内部/プログラム設計書
2	上司レビュー (レビュー形式：インスペクション) ※インスペクション：事前に定められた手順とチェックリストに従って第三者がレビューを行う、最も形式的なレビュー手法	・ プロジェクトの成果物とするために、所定の品質に達しているかの検証を目的として、各グループの上司に参加いただき実施する ・ レビュー形式は、成果物の作成者以外の第三者を参加させるインスペクションレビューとする	・ 外部設計書 ・ 内部/プログラム設計書
3	お客様レビュー	お客様要求事項が、外部設計書に反映されていることの確認を目的に、お客様と合同で実施する	・ 外部設計書

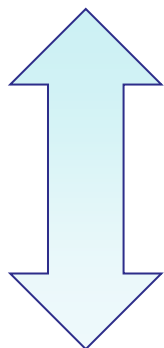
参考：レビューとは

レビューとは(2/2)

■ 問題指摘の優先度

以降の工程などに与える影響の大きさなどから、問題点を指摘する際に優先度を意識します。

優先度高



優先度低

- 漏れ：お客様と合意した要件、業務を遂行するために必要な機能、ユーザからの入力、ユーザへの出力など
- 曖昧さ：「複数の解釈が可能」「記述内容だけでは判断がつかない(説明不足)」など
- 誤り：意味の取り違いなどを起こさない軽微な誤字・脱字など

レビューアは、優先度の高いものから指摘するように心掛けます。

■ レビュー時の注意事項

- ・レビュー観点を意識すること

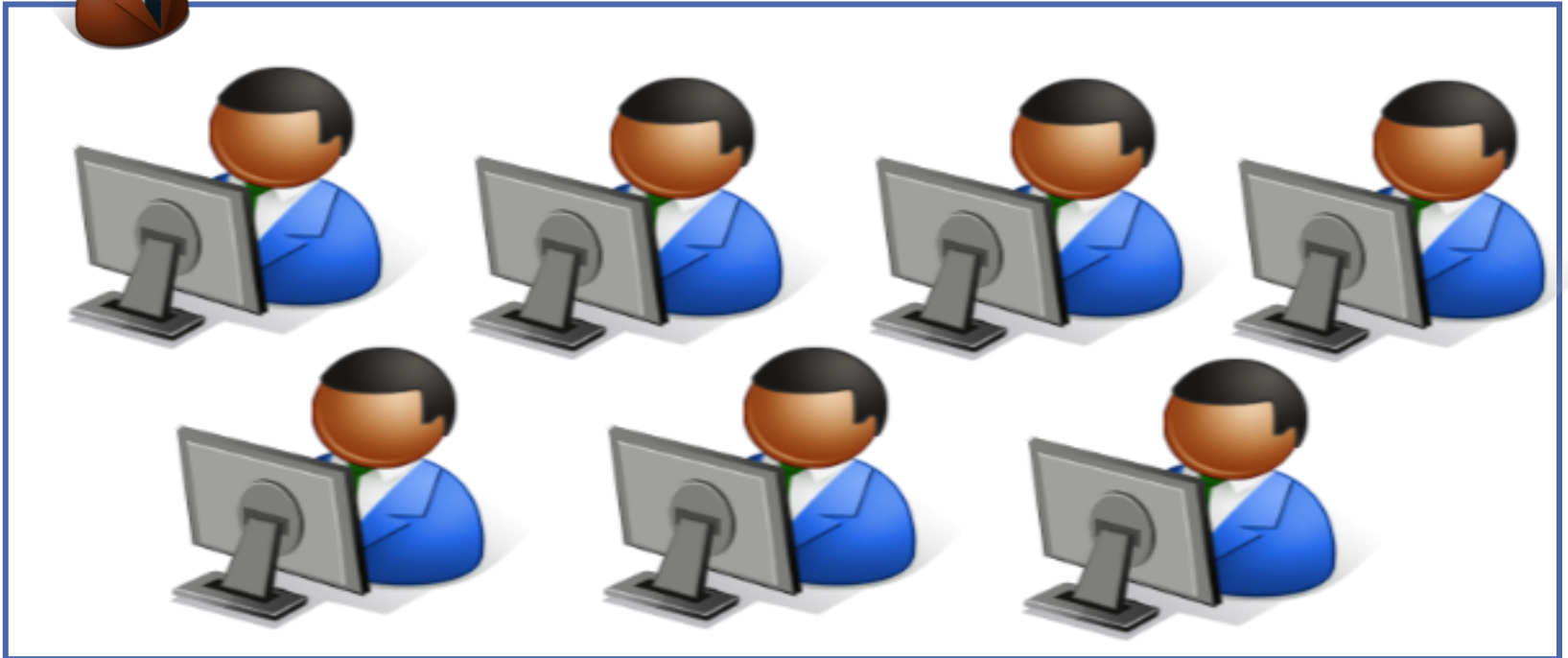
その際、開発に与える影響が大きいものから抽出することを意識する

- ・レビューアのコメントを批判しないこと

3.4 班編成とレビュー（お客様 / 上司）



お客様 / 上司役



- 個人ごとに演習を行います。
- お客様／上司役の講師がいますので、各種相談・レビューをしてください。

3.5 日々のレビュー



時間	作業
13:00～13:30	研修開始・事前説明など
13:30～14:00	レビュー①
14:00～14:30	レビュー②
14:30～15:00	レビュー③
15:00～15:20	休憩時間

時間	作業
15:20～15:50	レビュー④
15:50～16:20	レビュー⑤
16:20～16:50	レビュー⑥
16:50～17:20	レビュー⑦
17:20～17:30	講師からのアドバイス

- レビューでは以下の内容をお客様／上司役に報告してください。
 - ・進捗状況
 - ・現在の課題
 - ・懸念事項
- お客様／上司役からの質問もありますので、回答できるように準備をしてください。
- 時間は貴重です。効率よくレビューを進められるように準備をして臨みましょう。
⇒技術的な問題も1から聞くのではなく、まずは個人で解決策を考えてお客様／上司役に提案するように意識しましょう
- 演習ではなく、実際の業務のつもりでお客様／上司役と応対をしてください。

参考：試験密度/開発バグ密度とは

試験密度/開発バグ密度とは(1/3)

■ 試験密度

ソフトウェアの規模に対して、どの程度テストを実行しているかを示す指標

一般的に、「テスト項目数/システム規模」で算出する

テストケース数を設定する際の参考値として利用できるほか、テストの十分性の評価に活用することができる

◆ システム規模の尺度

- ・ FP(Function Point)...機能数をシステム規模と考える

- ・ SLOC (Source Lines Of Code) ...空行、コメント行を除くソース行数

※本開発演習では、ソースコード行数を元にシステム規模を算出する手法を採用

■ バグ密度

ソフトウェアの規模に対して、どの程度バグが発生しているかを示す指標

一般的に、「バグ数/ソフトウェア規模」で算出する

■ 指標の評価

試験密度やバグ密度は多すぎても少なすぎても良くない。どちらの値も、適正な範囲内に存在していることが重要である

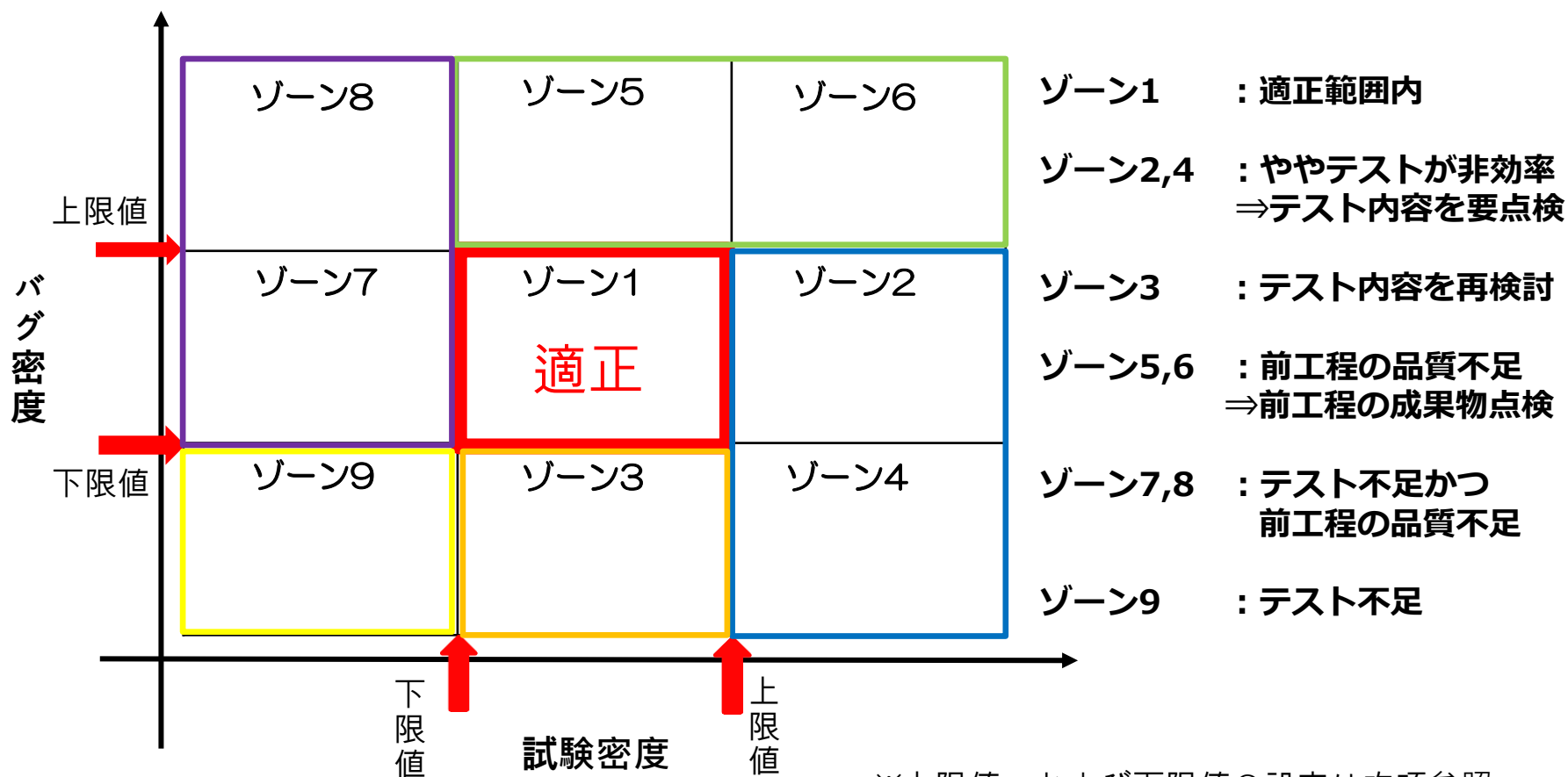
例えば、試験密度が高すぎると試験にかかるコスト（工数、費用）が膨大になり、バグ密度が低すぎると試験項目の抜け漏れが疑われる

各指標を見る際には、値が適正範囲内におさまっているかどうか確認する

参考：試験密度/開発バグ密度とは

試験密度/開発バグ密度とは(2/3)

- ゾーン分析
- 縦軸を「バグ密度」、横軸を「試験密度」として、各値が適正值かどうかを分析する品質管理手法の1つ



※上限値、および下限値の設定は次項参照

参考：試験密度/開発バグ密度とは

試験密度/開発バグ密度(3/3)

■ 上限値、下限値の決定方法

上限値、下限値の設定では、下記①または②のいずれかを選んで決定する。

①自社の蓄積データの活用

②客観性の高いデータの活用

(参考) : IPA 情報処理推進機構「ソフトウェア開発分析データ集2022」

<https://www.ipa.go.jp/ikc/reports/20220926.html>

●本演習における試験密度/開発バグ密度の参考値

	試験種別	-30%	基準値	+30%
試験密度 [項目/KL]	単体試験 (UT)	73.561	105.09	136.614
	結合試験 (IT)	34.164	48.81	63.447
	総合試験 (PT)	6.077	8.68	11.287
開発バグ密度 [件/KL]	単体試験 (UT)	3.43	4.89	6.36
	結合試験 (IT)	1.24	1.77	2.30
	総合試験 (PT)	0.31	0.44	0.57

※項目/KL : ソースコード1000行項目あたりの試験項目数
件/KL : ソースコード1000行項目あたりのバグ件数

4. システム開発演習 評価のポイント

評価のポイント(定量評価)

- 時間延長することなく、納品日を意識して時間管理できたかどうか
(プロセス:時間管理)
- 計画した通りにプロジェクト進行ができたかどうか
(プロセス:計画、進捗管理)
- ひとり1機能以上を開発し、個人の目標機能数を達成することができたかどうか
(製造工程:プログラム・個人)
- 個人の目標バグ数を達成できたかどうか
(テスト工程:品質管理・個人)

評価のポイント(定性評価)

- 開発標準を意識して進められたかどうか
(プロセス:開発標準)
- 週報や議事録で顧客や上司との情報共有ができたかどうか
(プロセス:情報共有)
- 役割を意識した取り組みをし、個人の役割に対する目標が達成できたかどうか
(プロセス:役割)
- 顧客が要望した通りに各工程のドキュメント作成ができたかどうか
(設計工程:ドキュメント)
- コーディング規約の通りにメンテナンス性を意識したプログラミングができたかどうか
(製造工程:プログラム)
- バグ管理簿を使い適切な項目消化、バグ摘出ができたかどうか
(テスト工程:品質管理)
- 任意機能や任意項目を実装できたか(特別評価)

5.1 成果発表会

ねらい

システム開発演習の成果を発表します。チーム単位で、電子商取引システムのデモを含むプレゼンテーションを実施します。

合わせて、相手に伝わる効果的な資料の作成方法と、相手に理解されやすい効果的なプレゼンテーション技法を習得します。

成果発表会の内容

- 発表者は、代表者 1 名
- 発表内容は、以下の通りとします。（時間は目安です）
 - システムの概要説明（1分）
 - スケジュールについて（1分）
 - テーマ：目標と実績、遅延した場合は原因と対策
 - ショッピング機能デモンストレーション（2分～3分）
 - 個人の成果（一人ひとり）（4分～5分）
 - テーマ：目標に対する成果、苦労した点、配属後に意識すること（1枚のスライドにまとめる）
 - チームの成果（1分）
 - テーマ：試験密度やバグ密度、苦労した点、配属後に意識すること