

ソフトウェア工学A チーム開発

鷺崎 弘宜

早稲田大学グローバルソフトウェアエンジニアリング研究所
国立情報学研究所 客員教授
エクスモーション 社外取締役



Twitter: @Hiro_Washi washizaki@waseda.jp

<http://www.washi.cs.waseda.ac.jp/>



目次

- アジャイル開発の復習
- アジャイル開発の導入条件と品質
- アジャイル品質パターン

アジャイル開発 補足



Nordstrom Innovation Lab “Flash Build”

<https://www.youtube.com/watch?v=2NFH3VC6LNs>

<https://www.youtube.com/watch?v=hNy2h6GG4ng>

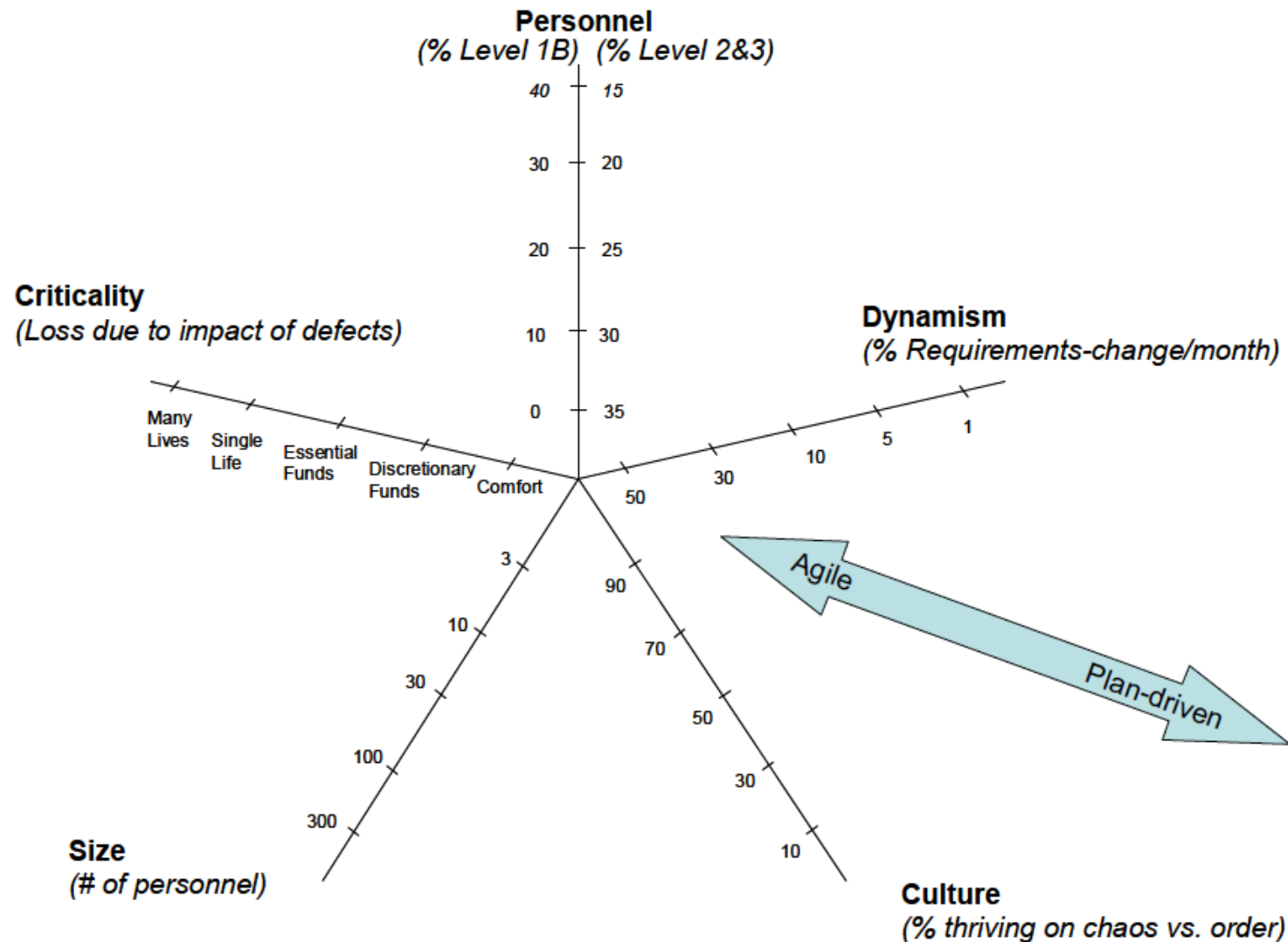
日本語 Subtitles by Joe Justice - Updated Jan 2021

Subtitles by Joe Justice

“Flash Build”の例におけるアジャイルプラクティスとは？

- 参考: アジャイル型開発におけるプラクティス活用 リファレンスガイド
<https://warp.da.ndl.go.jp/info:ndljp/pid/10128972/www.ipa.go.jp/sec/softwareengineering/reports/20130319.html>

アジャイル開発導入に向けた条件



Barry Boehm, Richard Turner, "Balancing agility and discipline: evaluating and integrating agile and plan-driven methods," Addison-Wesley, 2003.

The Aerospace Company, "Ground System Architectures Workshop", 2016.

レビュー



レビュー

- プロジェクト担当者、管理者、ユーザー、顧客、ユーザー代表、またはその他の関係者が、コメントや承認を得るためにソフトウェア製品を検査するプロセスまたは会議 [IEEE 1028] 。
 - より狭義には、ソフトウェアを読むことによる早期の欠陥や問題の検出
 - 開発のフィルター
 - 人間はエラーを起こしやすく、第三者は発見しやすい
- 利点
 - 50-60%の欠陥が設計中に挿入される傾向がある。
 - 形式的なレビューで最大75%程度まで発見可能
 - 理解、合意、教育



チェックリスト

- ソフトウェアの欠陥を発見するための一連の項目および/または手順
 - より短時間でより多くの欠陥を発見するため
 - 項目は通常、特性と目的によって分類
 - コーディング標準を含むこともある
- 項目の情報源
 - プログラミング言語やモデリング言語の仕様
 - 過去の開発で頻発した欠陥の種類
 - 経験的に検証されたグッドプラクティス
- 例
 - すべてのファイルは、開いた後に必ず閉じられるか？
 - 対象の名前はシンプルで適切か？



プロジェクト



目標と方法

- 目標

- UMLを用いたオブジェクト指向モデル中心のソフトウェア開発において、
- ドメイン分析から実装までの基本的な技法とツールを、小規模なプロジェクトを実施することで習得する。
- レビューによる開発資料の検証を行いながら開発プロセスを学ぶチーム開発の難しさとそれに対するソフトウェア工学の必要性を理解する。

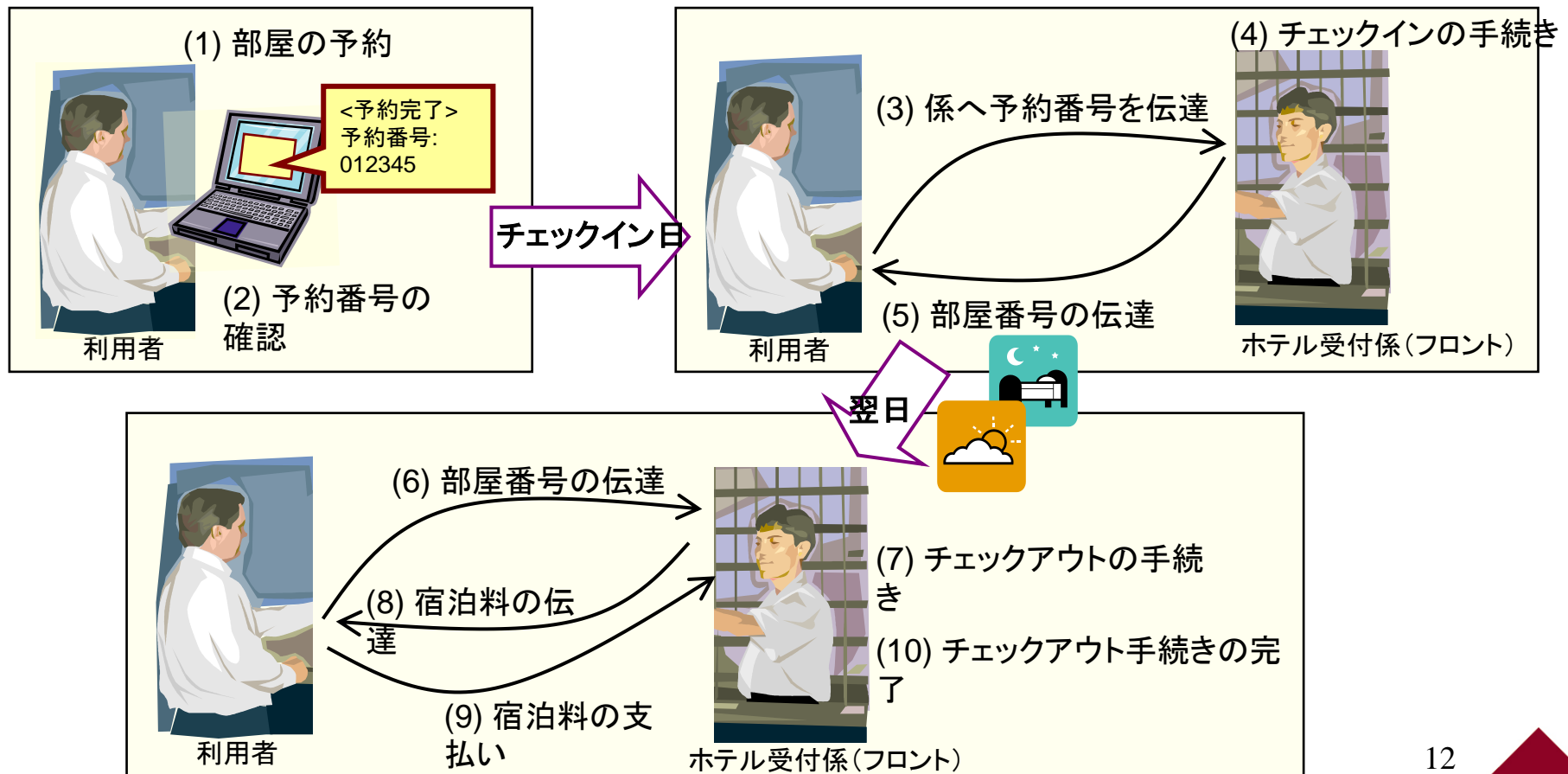
- 方法

- 講義中(および講義外)のチーム開発
- Githubによる開発管理: 今回の場合、ドメイン分析やユースケース単位の要求分析・システム分析といった形でIssueを作成して進捗管理および成果物管理するとよい
- 分析設計: すべて
- 実装: 提供される半完成コードの理解と一部の追加実装
- 最終レポート: 個人単位



対象: ホテル予約システム (HRS)

- ホテル予約システムのオブジェクト指向分析設計
- Java言語による部分実装



演習1: ドメイン分析

- HRSの問題領域を分析し、分析結果をUMLクラス図の形式で表現しなさい
- 具体的な状況に対して1つ以上のUMLオブジェクト図を作成し、クラス図を検証しなさい
- ダイアグラムをレビューし、必要に応じて修正しなさい



ドメイン分析・レビュー項目

対象	項目	○
全体	主要な概念および概念間の関係が網羅されている	
	対象とする問題領域の用語表・辞書として機能する	
	具体的な実装方法の推定とはなっていない	
	配置や色、フォントなどが読みやすい	
クラス	主に名詞・名詞句(複数の単語の組み合わせによる名詞と同等のまとまり)・代名詞が、クラス・属性・属性として挙げられている	
	直観的で明確なクラス名がつけられている	
関係	全ての関係について、両端を用いて違和感なく読める	
	汎化・集約の関係について、必然性がある	
	必要に応じて多重度・ロールが用いられている	

Ex2: 要求分析

- HRSに期待される重要な動作(機能)を3つ以上特定し、UMLのユースケース図の形で表現しなさい
- 各ユースケースの振る舞いを、ユースケース記述ならびにアクティビティ図として記述しなさい
- 図と記述をレビューし、必要であれば修正しなさい



要求分析・レビュー項目

対象	項目	○
全体	要求される機能、関係するアクタを網羅している	
	抽象的な機能要求の記述ではなく、具体的な振る舞いの記述となっている	
	配置や色、フォントなどが読みやすい	
ユースケース	各ユースケースはまとまった振る舞いの単位である	
	現在時制かつ能動態の動詞句としてユースケース名がつけられている	
	全てのユースケースの抽象度はおおむね同程度である	
アクタ	関わるユースケースとのみ関係づけられている	
ユースケース記述	必要な事前条件、事後条件、基本系列、代替・例外系列を明確かつ簡潔に記述している	
	システムへの入力に加えて、システムからの応答を明記している	
	各ステップは現在時制と能動句で記述されている	

Ex3: システム分析

- システム(ユースケース)分析を実施し、分析結果をUMLコラボレーション図とUMLクラス図の形式で表現しなさい
- その過程で、ドメイン分析結果と要求分析結果を修正しなさい
- ダイアグラムをレビューし、必要に応じて修正しなさい



システム分析・レビュー項目

対象	項目	○
全体	ユースケースが要求する全ての振る舞いを網羅している	
	必要なデータとその置きどころが明確化されている	
	コラボレーション図における相互作用や可能なオブジェクト数が、クラス図の関係・操作・多重度と対応している	
オブジェクト	バウンダリ、コントロール、エンティティの3つ組が、それぞれ当該種別の役割に合致して識別されている	
	コントロールはユースケース当たり5個以内である	
相互作用	ユースケースの基本系列に加えて、代替・例外系列も描かれている	
	描かれた相互作用とその順序が、対応するユースケース記述に合致している	
	バウンダリ、コントロール、エンティティ間で書いてよい相互作用のみが書かれている	