NaITE#23 開催レポート

2017 年 7 月 27 日

報告者：角田　俊、藤沢　耕助（NaITE）

1 開催概要

2017 年 7 月 16 日　NaITE#23 を以下のように開催しました。

開催日時	2017/7/26（日） 13:30 ～16:45		
開催場所	豊洲文化センター		
メインテーマ	Scrum 入門＆Agile Japan 2017 長崎サテライト参加報告		
キーワード・タグ	Agile Scrum Git		
プログラム	13:30～13:40	オープニング「開会のご挨拶」	NaITE 運営スタッフ
	13:40～14:10	セッション 1「Agile Japan 2017 長崎サテライト参加報告」	佐藤　博之氏
	14:10～14:25	セッション 2「Git Flow を運用するために」	角田　俊氏
	14:35～16:35	セッション 3「SCRUM 入門 -脱・やってるつもり SCRUM-」	朱峰　錦司氏
	16:35～16:45	クロージング	NaITE 運営スタッフ

©NaITE All rights reserved.

2 勉強会の様子

今回は「Scrum 入門＆Agile Japan 2017 長崎サテライト参加報告」というタイトルで勉強会を行いました。

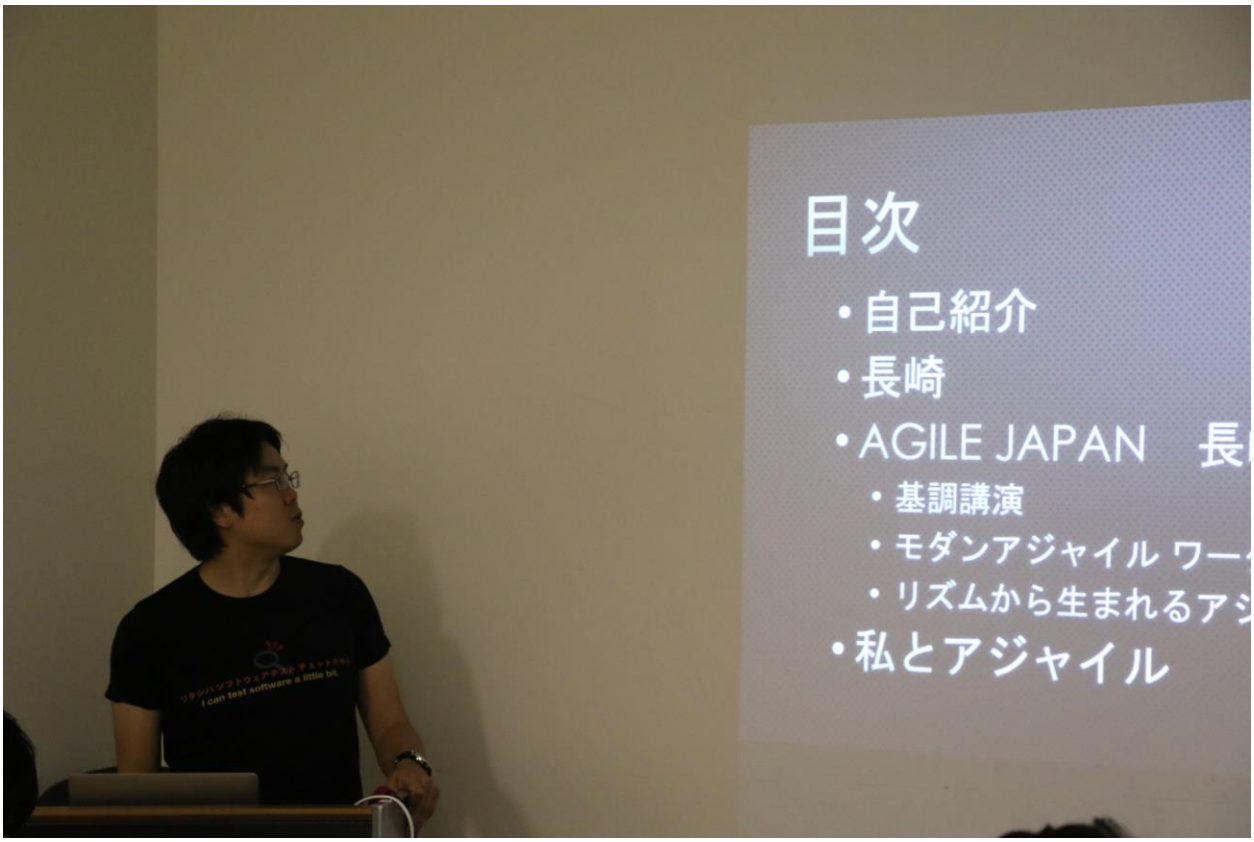
7 月 1 日に開催された Agile Japan 2017 長崎サテライト with NaITE の参加報告を佐藤 博之氏にして頂きました。

また、もう一つのテーマとして、朱峰 錦司氏に Scrum を紹介して頂きました。Scrum は、ソフトウェアのアジャイル開発手法として知名度の高い開発手法のフレームワークです。Scrum を業務で実践し、社内では Scrum の教育に取り組まれている朱峰氏に Scrum の根本にある考え方や、普段どのように実践しているのかを紹介して頂きました。

2.1 オープニング

オープニングでは NaITE スタッフから本イベントの注意事項、NaITE についての紹介、当日のプログラムについて説明をいたしました。その他現在活動中の SIG、活動を完了した SIG についての案内も実施いたしました。

2.2 セッション 1「Agile Japan 2017 長崎サテライト参加報告」



©NaITE All rights reserved.

佐藤 博之氏から Agile Japan 2017 長崎サテライト参加報告を行って頂きました。佐藤氏は東京で行われた Agile Japan 2017 にも参加しており、更に長崎で行われたサテライトにも参加していました。両方に参加して、どの様に感じたのかを報告して頂きました。

最初に、アジャイルソフトウェア開発宣言を紹介してアジャイルの概要を説明して頂き、Agile Japan 2017 長崎サテライト参加報告で行われた以下 3 つのセッションについて紹介して頂きました。

- Agile Japan2017 の基調講演で紹介されたモダンアジャイル
- モダンアジャイルのワークショップ
- リズムから生まれるアジャイルな開発

アジャイル開発でよくある誤解として、アジャイルソフトウェア開発宣言の文字が強調されている右側ばかりが重視されるようになってしまったが、アジャイルソフトウェア開発宣言には左側も重要だと書かれており、プロセスやドキュメントを軽視するのは間違いであるという紹介がありました。

参考：アジャイルソフトウェア開発宣言：http://agilemanifesto.org/iso/ja/manifesto.html

当日の質疑応答について、以下に記載します。

■当日の質疑応答

Q1. モダンアジャイルの 4 つの原則はどういう経緯で生まれたのか？

A1. アジャイルに対する誤解を払拭しきれないところがあった。そのため、新しく「モダンアジャイル」として定義した。モダンアジャイルを提唱した背景については、長崎サテライトの関氏の発表資料で詳しく説明されている。

参考：モダンアジャイルワークショップ - Agile Japan 2017 地方サテライト版

https://www.slideshare.net/fullvirtue/agile-japan-2017-agilejapan-76948191

Q2. 安全の原則は、プロダクトのセーフティとは異なるのか？

A2. プロダクトのセーフティも含む。何かのトラブルが起きた時に、誰かが責められる状況になるのは安全ではない。トラブルを起こした人が悪いのではなく、トラブルを起こしてしまえる環境が悪いと考える。

©NaITE All rights reserved.

2.3　セッション２「Git Flow を運用するために」



「Git Flow を運用するために」と題して、角田 俊氏に発表いただきました。

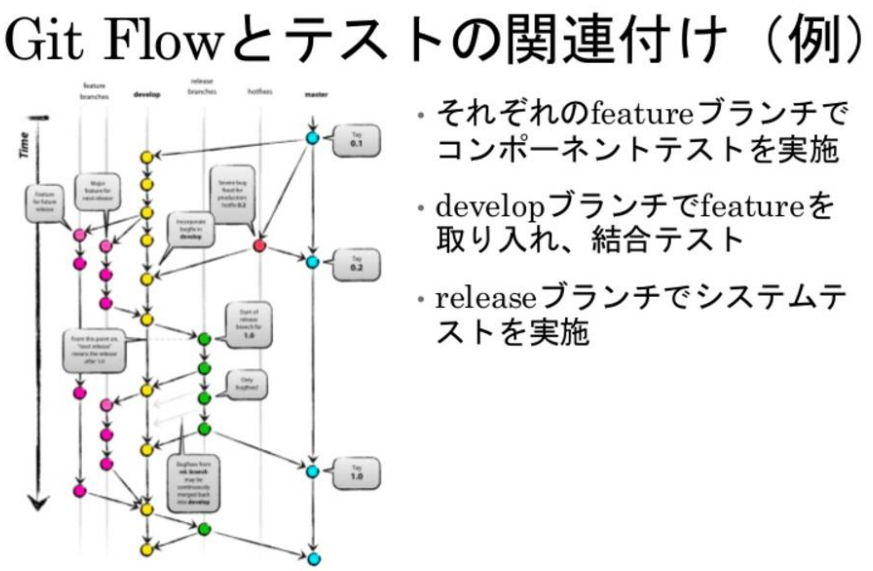
Git の導入にあたり、特にルールもなくブランチを使っていると無秩序なブランチの作成やマージが行われてしまうことがあります。こうした問題を解決するために、「ブランチモデル」というブランチの管理方法がいくつか考案されました。Git Flow は、そのブランチモデルの一つです。

角田氏は、Git Flow 導入のメリットとして以下の点を挙げられました。

- 機能の独立したリポジトリが作成できる
- 機能をリリースする、しないの判断が後から選択可能
- 開発とテスト、テスト工程による明確なブランチ分割
- アジャイル開発や、イテレーション開発に向いている

©NaITE All rights reserved.

角田氏は、Git Flow で作成するブランチとテストの関連付けについて図を示しながら解説されました。



角田氏の発表資料「Git Flow を運用するために」p.9

実際に現場に Git Flow を導入する際は、現場によって定義しているテストレベルが異なる点に気をつけると良いと説明されました。例えば JSTQB で定義しているコンポーネントテストなどのテストレベルの他にも、実際の現場では「既存システムとの統合テスト」「他社システムを含めたシステムテスト」などプロジェクト独自の定義をしている場合があります。そのため、Git Flow を運用する前に自分たちのテストレベル、テストフェーズに合ったブランチの数や役割を定義する必要があると説明されました。

また、Git Flow の Feature 分割に失敗した経験を紹介いただき、Git Flow を効率良く運用するには適切な Feature 分割が必要であること、Feature が独立した検証可能な単位の最小構成である必要があることを強調されました。

参加者の方からは実際の Git 運用の経験を踏まえた意見が寄せられ、活発なディスカッションとなりました。

なお、以前 NaITE で開催した NaITE#13「Docker 入門 & Git 運用のコツ」においても GIt 運用について取り上げています。こちらも合わせてご参照ください。

参考：NaITE#13「Docker 入門 & Git 運用のコツ」開催レポート

http://naite.swquality.jp/?p=154

©NaITE All rights reserved.

2.4　セッション 3「SCRUM 入門 -脱・やってるつもり SCRUM-」



普段から業務でアジャイルを実践されており、アジャイルに関する社内教育も行われている朱峰氏に Scrum について紹介して頂きました。

最初に、最近では「なんちゃってアジャイル」を実施している現場が多いと感じており、「アジャイルとは開発のやり方ではなくビジネスのやり方である」というのが朱峰氏の考え方であると説明がありました。一般的に、プロダクトづくりの考え方として以下の二つがあると紹介されました。

- プロダクトアウト
  - 物を作ってから売り方を考える
- マーケットイン
  - 売れるものを調査して物を作る

今はビジネスに求められるスピードが速くなっており、プロダクトアウト、マーケットイン、どちらかだけでは通用しない時代になってきており、両方の考え方を取り入れる必要があると説明されました。そのため、物を速く作り、そのフィードバックを元に改善を繰り返してプロダクトを作っていかなければいけないとのことでした。時代に迅速に対応するために、アジャイルな開発手法が注目されているということでした。また、”Complicate”と”Complex”の

©NaITE All rights reserved.

違いについても触れ、”Complicate”は解が一つ存在するもの、”Complex”は解が存在するとは限らないものであり、正しいことの検証が出来ないものであるという紹介がありました。アジャイル開発で向いているプロダクトは”Complex”なプロダクトであると説明されました。アジャイル開発では、高速に開発して、それを元として高速にフィードバックを得ながら開発していくスタイルだということも出来ます。

そんなアジャイル開発を実現する方法のプロセスフレームワークとして作成されたのが Scrum です。Scrum はプロセスフレームワークであり、技術的な要素については XP（エクストリームプログラミング）等、他の技術や手法で補完する必要があるということでした。

スクラムの基本は経験的プロセス制御であり、自分達で反復的かつ漸進的アプローチでチームを成長させていくことであり、それを実現するために以下の３つの概念があるということでした。

- 透明性
  - 見える化、共通理解など
- 検査
  - 成果物、進捗を定期的に確認する
- 適応
  - 不備があれば改善する
  - リーダーが指示するのではなく、メンバーが自律的に実施する

また、Scrum には以下の要素があるということで、それぞれの役割や内容などを紹介して頂きました。

- ロール
  - プロダクトオーナー
  - 開発チーム
  - スクラムマスター
- イベント
  - スプリント計画
  - デイリースクラム
  - スプリントレビュー
  - 振り返り
- 成果物
  - プロダクトバックログ
  - スプリントバックログ
  - インクリメント（成果物）

©NaITE All rights reserved.

実際に Scrum のプラクティスとして、以下のものを紹介して頂きました。

- 相対見積もりとプランニングポーカー
- バーンダウンチャート
- Scrum ボード（カンバン）

当日の質疑応答について、以下に記載します。

■当日の質疑応答

Q1. もし、スプリント計画を立てる日にメンバーが休んでしまった場合はどうするのか？

A1. 決められた曜日、時刻にミーティングを行うことが大切となる。そのため、ミーティングの予定はできるだけ変更せずに実施し、休んだメンバーがいた場合には後日伝える様にする。

Q2.Scrum ボード上で完了したタスクが、後で完全に完了していないことが発覚した場合はタスクを新しく作るべきか？それとも、タスクを「完了」から戻すべきか？

A2.対策が必要なのが認識できることが大事であるため、どちらでも良い。

3 参考情報・リンク等

当日の発表資料等は以下をご参照ください。

・OP 資料

https://www.slideshare.net/NaITE_Official/naite-23scrumagile-japan-2017

・Git Flow を運用するために

https://www.slideshare.net/ShunTsunoda/git-flow-77929819

©NaITE All rights reserved.

4 次回の予定等

次回は 9/17　NaITE ＃24「テストカタマリー　ワークショップ＆解説」として開催いたします。

https://nagasaki-it-engineers.connpass.com/event/63088/

以上

©NaITE All rights reserved.