

みんなで Scrum演習!!!

細令 晋太郎



©2015 Shintaro Hosoai



演習概要

- GR-SAKURAを使った簡易なソフトウェア開発を例題にスクラムフレームワークの流れを実習する

- お題：近接表示システム

対象までの距離をLEDで表示

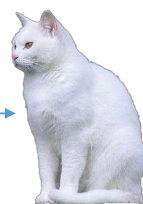
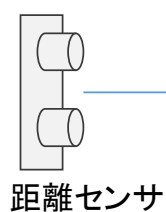
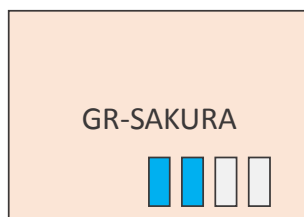
100cm～：消灯

75cm～：1つ点灯

50cm～：2つ点灯

25cm～：3つ点灯

～25cm：4つ点灯



©2015 Shintaro Hosoai

演習概要：実装詳細（1）

- 下記3つのクラスで構成します

Sketch.cpp
void setup() void loop()

setup():LEDピンの初期化
各インスタンス生成
loop(): 距離取得, LED点灯

LED.cpp
void lightLED(int N)

lightLED(int N):
引数N個LEDを
点灯させる

Sonar.cpp
int getDistance()

int getDistance():
距離に応じたLED点灯数
を返す.
100cm~ : 0
75cm~ : 1
50cm~ : 2
25cm~ : 3
~25cm : 4



©2015 Shintaro Hosoi

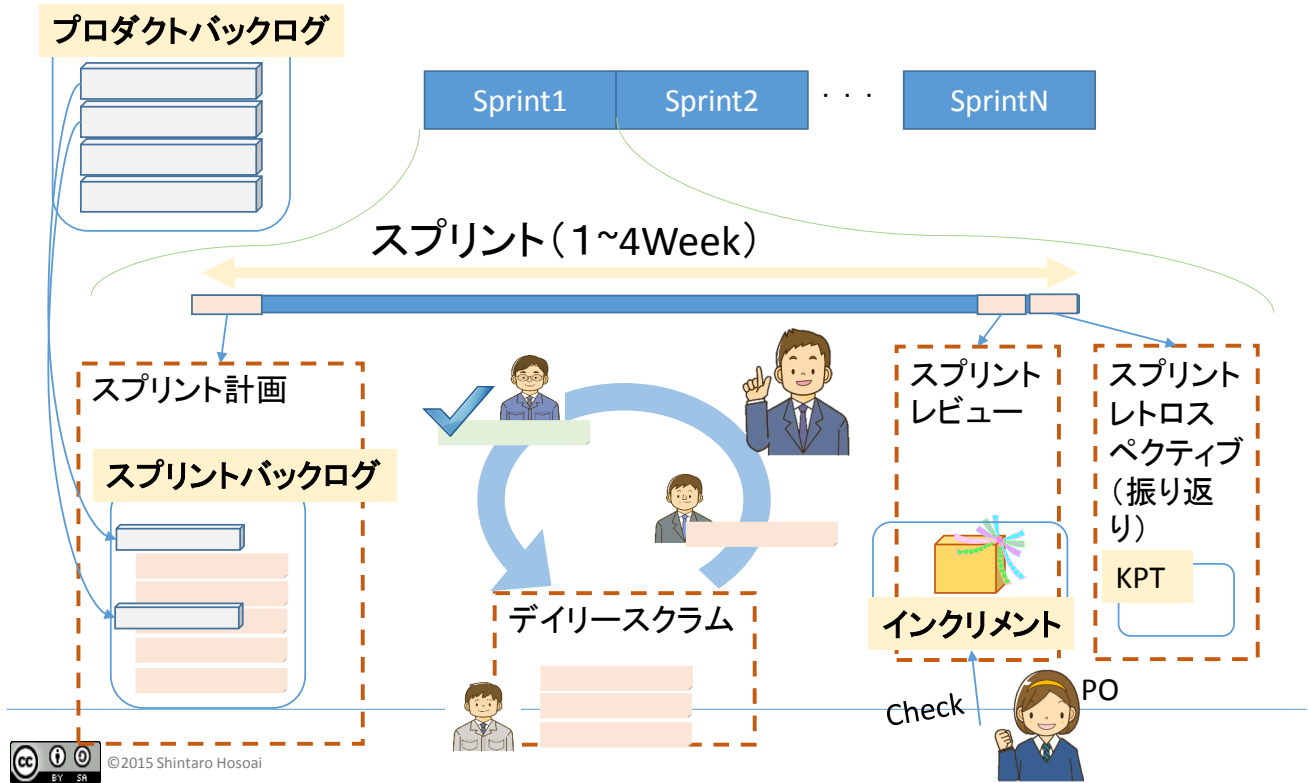
演習概要：実装詳細（2）

- LED点灯の方法
 - setup()内
 - pinMode(LED_PIN0, OUTPUT); // LED_PIN0-3まで同様に記述
 - 点灯
 - digitalWrite(LED_PIN0, HIGH);
 - 消灯
 - digitalWrite(LED_PIN0, LOW);
- SonicSensorクラス
 - SonicSensor(int trig, int echo, float coefficient = 1.0924);
 - コンストラクタ
 - SonicSensor sonic = SonicSensor(2,3);
 - float GetSonicSensor(void);
 - 距離取得
 - float distance = sonic.GetSonicSensor();



©2015 Shintaro Hosoi

Scrumフレームワーク



ルールを決める

- スクラムマスター
 - 各ミーティングの進行役
- プロダクトオーナー
 - DoDの確認やレビューで要求を満たしているか確認
- デベロッパー
 - 実装

今回はスクラムマスター, プロダクトオーナー
はデベロッパーも兼任してください
スプリント1, 2で役割を交代してください

プロダクトバックログ

Product Back Log

価値: 見積:
LED表示

価値: 見積:
距離のLED表示

option

価値: 見積:
バイナリ値表示

- プロジェクト全体で作るべきものを見る化する
- 各項目はデモ（評価）が出来るものとする

- 今回は、項目は予め提供します。見積りと価値のパラメータを相対見積もりしてください



©2015 Shintaro Hosoi

プロダクトバックログ（2） 付箋例

価値:

★~★★★★★

見積り:

1, 2, 3, 5, 8, 13, ∞

★★★

5

項目名: 大きな字で分かりやすく簡潔に

デモ定義: 何が出来たら、この項目を完了と見なすか。



©2015 Shintaro Hosoi

プロダクトバックログの作成（2）：見積もり

- 各項目の「価値」と「難度」の見積もりを行う
- プランニングポーカー
- 1,2,3,5,8,13,∞,?といったカードで相対見積もりを行う。
- 一番簡単なタスクがどれか探す。そのタスクを2として、相対見積もりを行う。
- 価値は、その項目を満たした際の価値を見積もる。（優先順位の指標）。加えて依存関係も考える



©2015 Shintaro Hosoai

プロダクトバックログの作成（3）：ソート

- 価値と難度，依存関係に応じて，プロダクトバックログ項目のソートを行う。
- 常にプロダクトバックログ項目は修正する。
- スプリント計画毎，スプリント内でも必要があればミーティングを開き項目の追加や優先度の修正を行う。



©2015 Shintaro Hosoai

スプリントを開始する

1	スプリント計画ミーティング
2	
3	
4	
5	
6	デイリースクラム
7	実装
8	
9	
10	
11	デイリースクラム
12	実装
13	
14	
15	
16	スプリントレビュー
17	
18	振り返り
19	
20	

- スプリント計画ミーティング（5分）
 - プロダクトバックログからスプリントバックログを作成
- デイリースクラム（～1分）× 2
 - 進捗報告, タスク担当の決定
- 実装（8分）
 - タスクの実装
- スプリントレビュー（2分）
 - DoDの確認, プロダクトバックログ項目の達成確認
- 振り返り（3分）
 - KPT

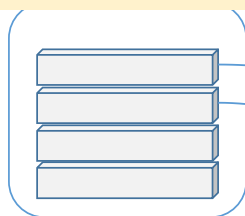


©2015 Shintaro Hosoi

プロダクトバックログからスプリントバックログへ

- スプリント計画ミーティングでは, そのスプリントで作業するプロダクトバックログ項目の選択と, 項目を見積もり可能なタスクに分割する.

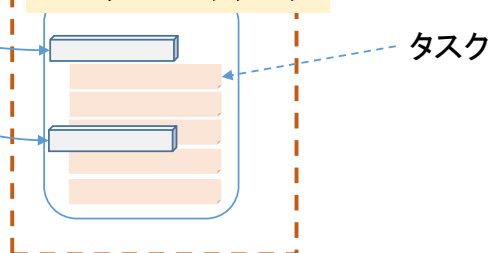
プロダクトバックログ



基本的にPBLは優先順位で並んでいるはずなので, 上から順に取り出すだけで良い.

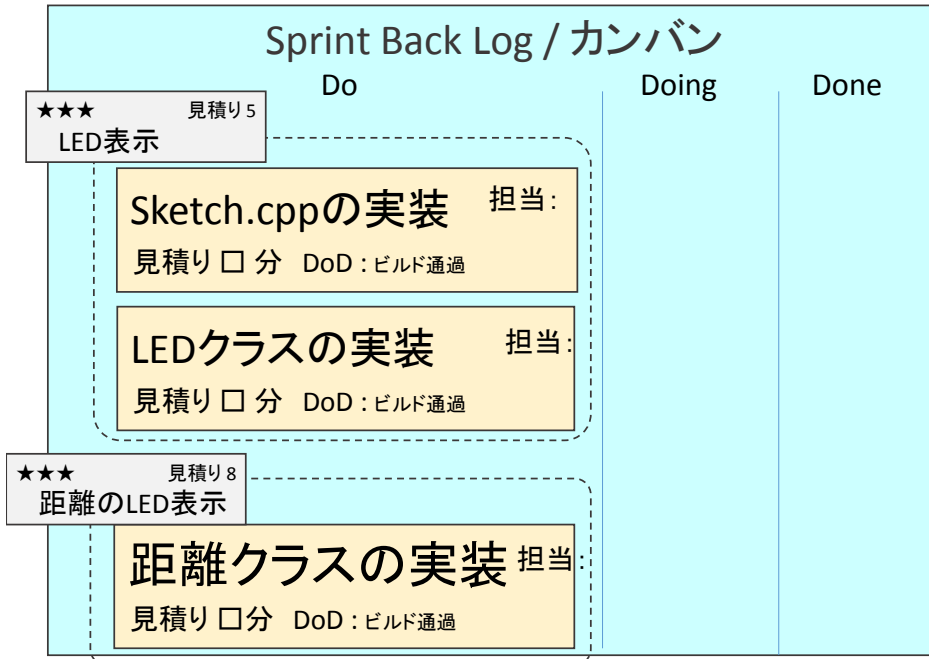
スプリント計画

スプリントバックログ



©2015 Shintaro Hosoi

スプリントバックログ

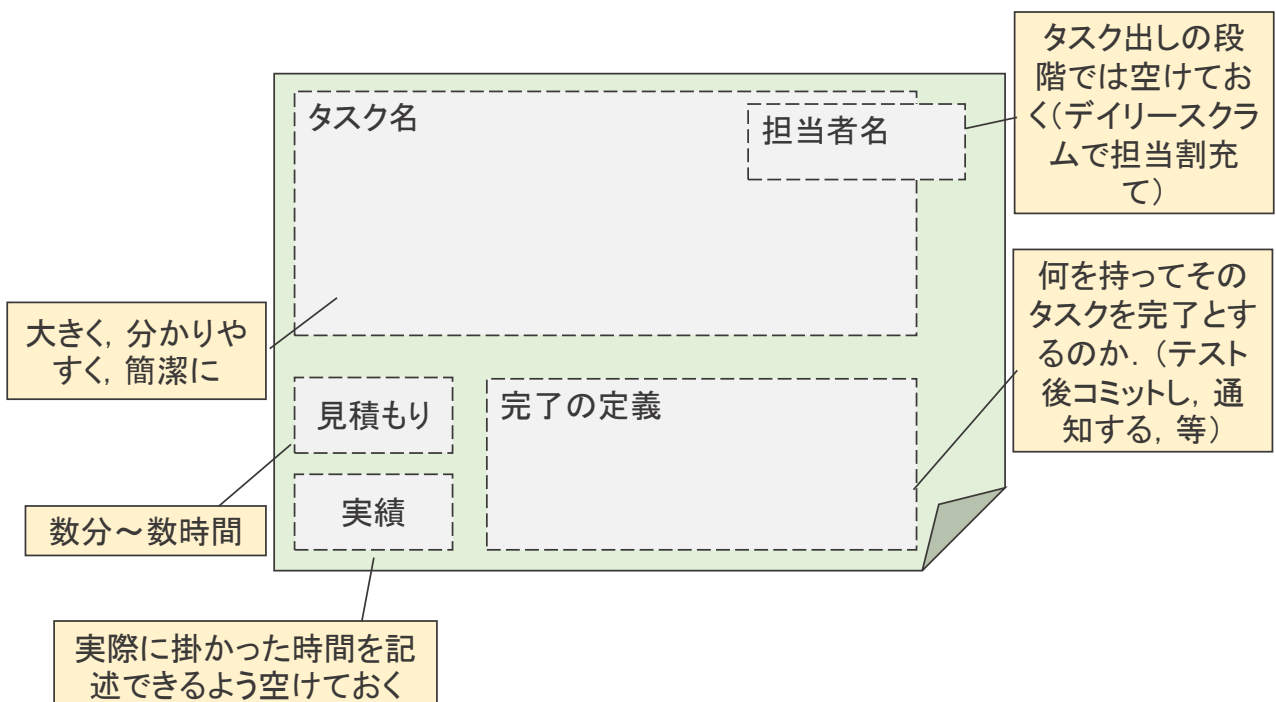


- 今回は予め項目を提供します.
- 時間の見積りを行ってください



©2015 Shintaro Hosoai

スプリントバックログ（2）：付箋例



©2015 Shintaro Hosoai

デイリースクラム & 実装

- 前回から今までに行ったこと, 状況
 - 完了したタスク, 現在作業中のタスクを申告
- 次に行うこと
 - 次に取り掛かるタスクを宣言
 - スプリントバックログのタスクに担当を記入, カンバンのDoingへ
- 問題点
 - 現在作業中のタスクで問題があれば報告
 - この場では問題解決は行わない
- 実装
 - 作業中のタスクが終わった場合はDoDを確認し, 実際に掛かった時間を記載しタスクをDoneに移動させる.
 - 次に取り掛かるタスクを決め担当を記入し, Doingへ
 - 作業中は, 個々の作業に集中しすぎず, 積極的に助け合う



スプリントレビュー

- プロダクトオーナーが中心となり下記を確認
 - スプリントバックログのタスクがDoDを満たしているか再度確認
 - プロダクトバックログの項目が完了している場合は, その項目の要求が満たされているか確認
 - 満たしていない項目や, 問題点があれば, タスクとしてスプリントバックログに追加



Sprint1 作業開始！



©2015 Shintaro Hosoai

LEDCAMP3

振り返り (KPT)

- 今回のスプリントを振り返り下記を洗い出す
- 各自が各項目を書き出し(2分), 全員で共有 (1分)
 - Keep :
 - 実施してよかったこと
 - Problem :
 - 問題点
 - Try :
 - 問題点に対する解決策
 - 新たに実施してみたいこと



©2015 Shintaro Hosoai

Sprint1 振り返り

- 今回の
- タスクの割当ては適切だったか？
- タスクの見積もりは合っていたか？
 - ずれていた場合、何を見過ごしていたか、どうすればより正確に見積もれるか
- チームをより効率よく回すにはどうすればよいのか？



©2015 Shintaro Hosoai

振り返り

- KPT法で以下の項目について、それぞれKPTを考える
 - 人、関係、プロセス、ツール

Keep	Try
効果があったため、引き続き行うこと	次回から挑戦すること
Problem	
問題があったため、改善すべきこと	

Keepはチームの実績のあるノウハウ
どんどん蓄積しよう



©2015 Shintaro Hosoai

Sprint2 作業開始！



©2015 Shintaro Hosoai



Sprint2振り返り

- 今回の演習を通じて、スクラムや開発の難しかった点を洗い出す
- チーム開発演習で実施するとよいと思ったことをKeep / Tryとして書き留めておく



©2015 Shintaro Hosoai

チーム開発演習に向けて

- 今回は非常に短い時間でスクラムのフレームワークを実践した。実際にはより長いタイムボックスで実施する、息切れしないペースをチームで作ろう
- チーム開発演習では、1スプリントを90分で実施する。スプリント内のタイムボックスはチームに委ねる
- プロダクトバックログ、スプリントバックログの項目を提供したが、演習中はチームで作成する
- 常にチームがより良く動けるにはどうすればよいか考え、共有してください