



How to use Redmine



Table of contents

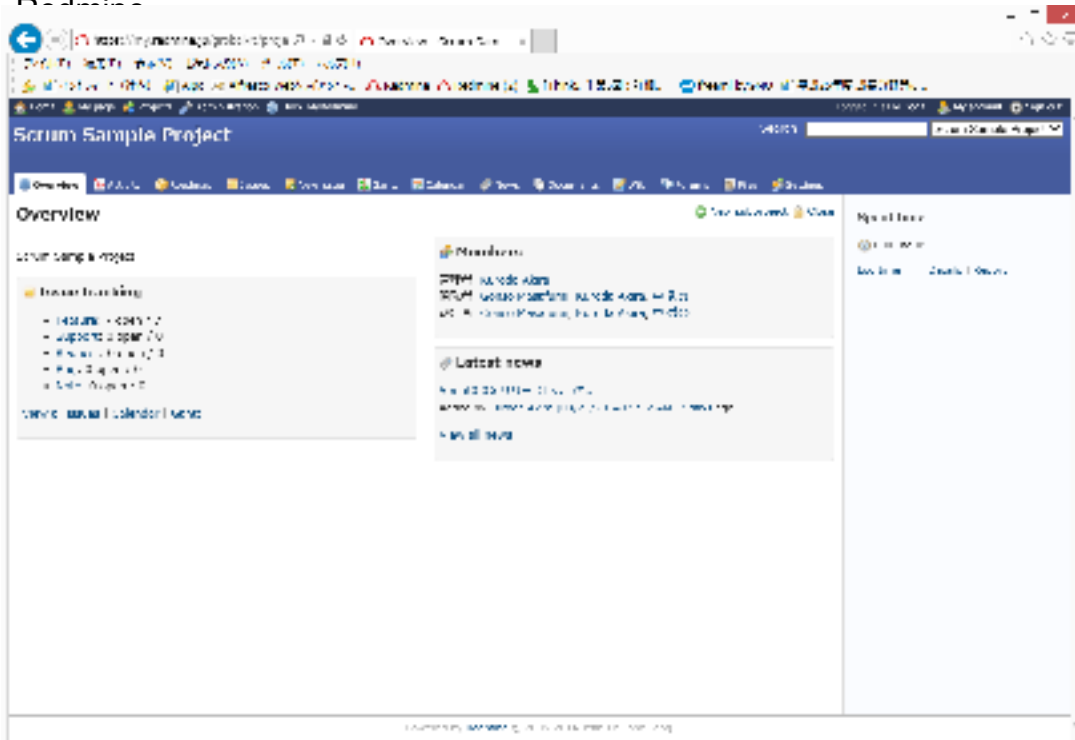
1. Summary of Redmine	3
2. Function summary of Redmine	4
3. Project organization diagram	5
4. Role in the project	6
5. Scrum framework	7
6. Ticket operative cycle	8
7. Trackers (Ticket kind)	9
8 Workflow of a tracker	
8.1 Workflow of a tracker (Request • Feature)	10
8.2 Workflow of a tracker (Issue)	11
8.3 Workflow of a tracker (Bug)	12
8.4 Workflow of a tracker (Support)	13
9. Operations Flow of Redmine	14
9.1 Sprint planning (Product backlog)	15
9.2 Sprint planning (Sprint backlog)	16
9.3 Sprint (Ticket is registered and updated)	17
9.4 Sprint (Daily scrum)	18
9.5 Sprint (Progress management)	19
9.6 Sprint review	20
9.7 Sprint release	21
9.8 Sprint retrospective (KPT)	22
9.9 Sprint retrospective (KPT) Information sharing	23
9.10 The solution of a Issue	24
10. Information for the exercise course	
10.1 Project information for the exercise course	25
10.2 User information for the exercise course	26
10.3 User information modification method	27

1. Summary of Redmine

Redmine is a project management system.

Redmine can perform management of a task.

This data explains the employment method for performing Surum development by Redmine.



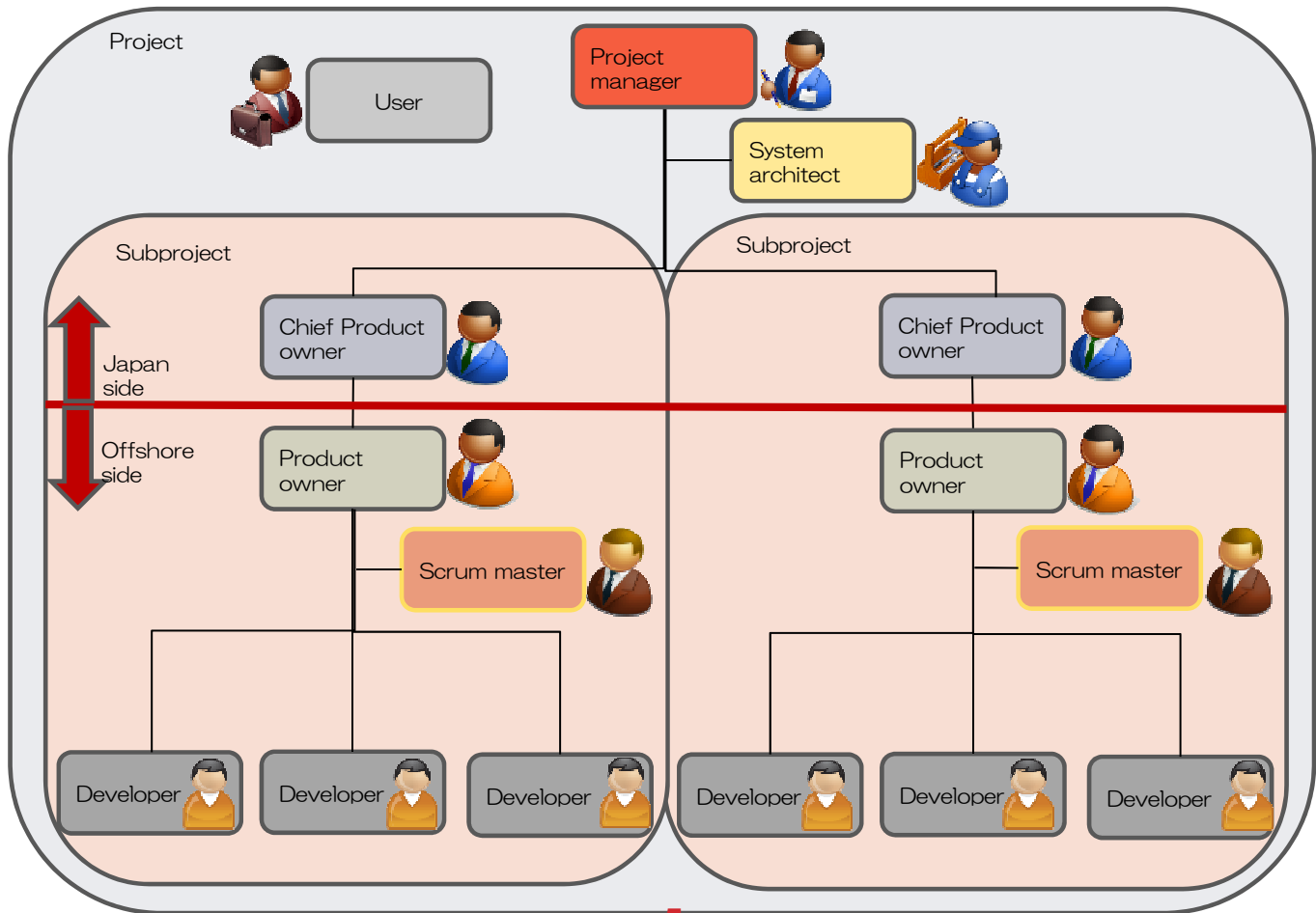
3

2. Function summary of Redmine

Function	Summary
Overview	The summary of the information on the whole project is displayed.
Activity	Record by the time series of all the work in a project is displayed.
Roadmap	Progress and lists of tickets for each of the sprint of the project is displayed.
Issues	Is a core feature. Create a task as a ticket.
Gantt	Progress of a ticket is displayed on a Gantt chart.
Calendar	A ticket and a sprint are displayed on a calendar.
News	It is a function which can publish the message to a project member. Used to product release announcements, etc.
Documents	It is a function which publishes the file shared within a project. Used to store building steps, such as.
Wiki	It is a function which can add and update the page for information disclosure. Used to share notes, such as Scrum team.
Forums	It is the function to argue about specific subject within a project. It is used in order to argue about a subject.
Files	It is a function which stores the file etc. which were released and it downloads within a team. It is used when releasing a product.

4

3. Project organization diagram



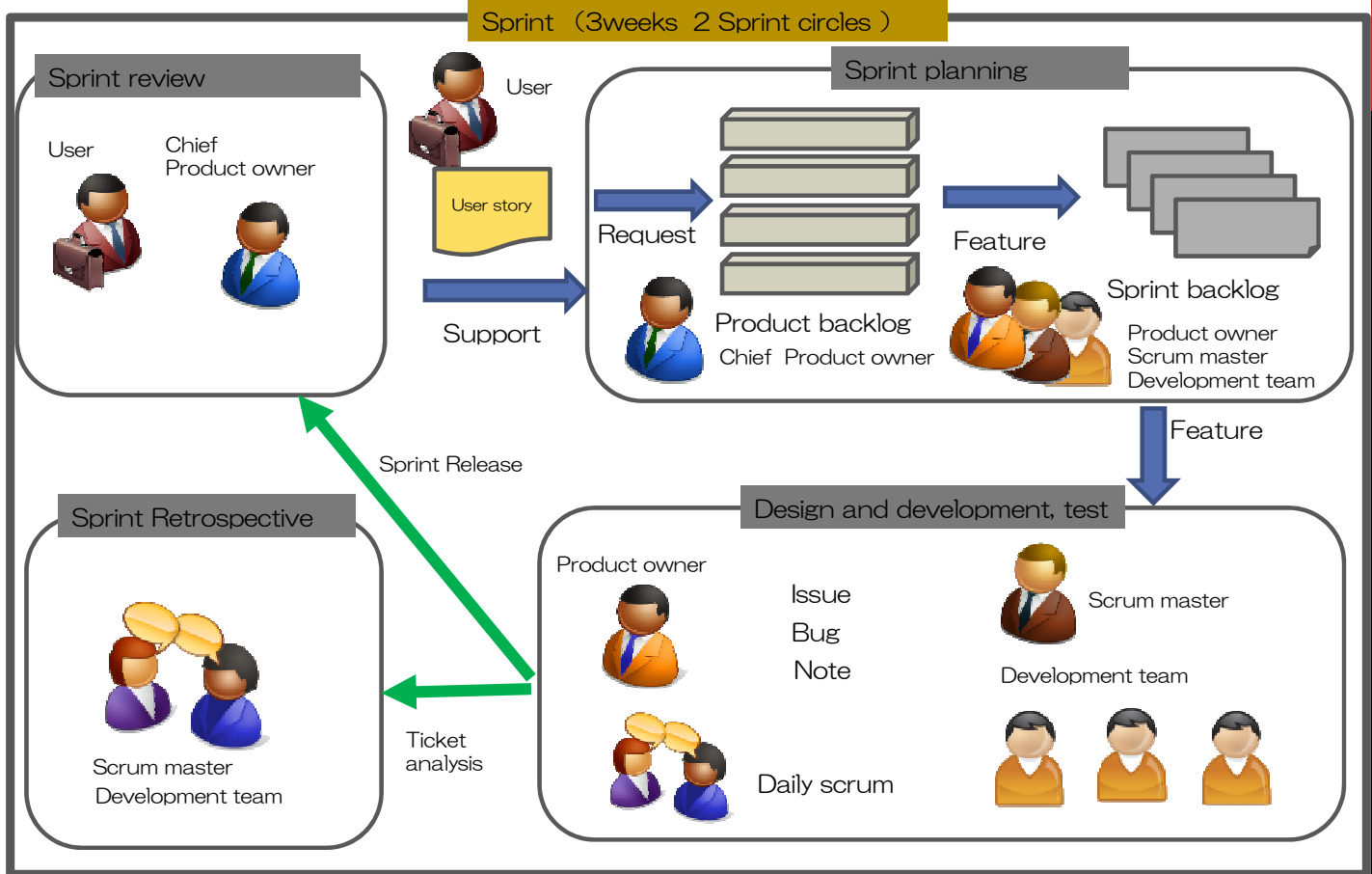
5

4. Role in the project

Role	Japan	Offshore	Summary
Project manager 	○		The person who summarizes the whole project
System architect 	○		The person who copes by a technical problem The person who supports a chief product owner chief
User 	○		The person who shows a user story (demand matter) of the application.
Chief Product owner 	○		The person who decides specifications of the application
Product owner 		○	The person who refers to a chief product owner for a problem of the application I perform the judgment of the sprint release.
Scrum master 		○	The person who checks whether sprint back log is updated.I consider it so that the efficiency of the team becomes biggest Person.
Development team 		○	The person who updates the sprint back log I have the ability of implementation and the test.

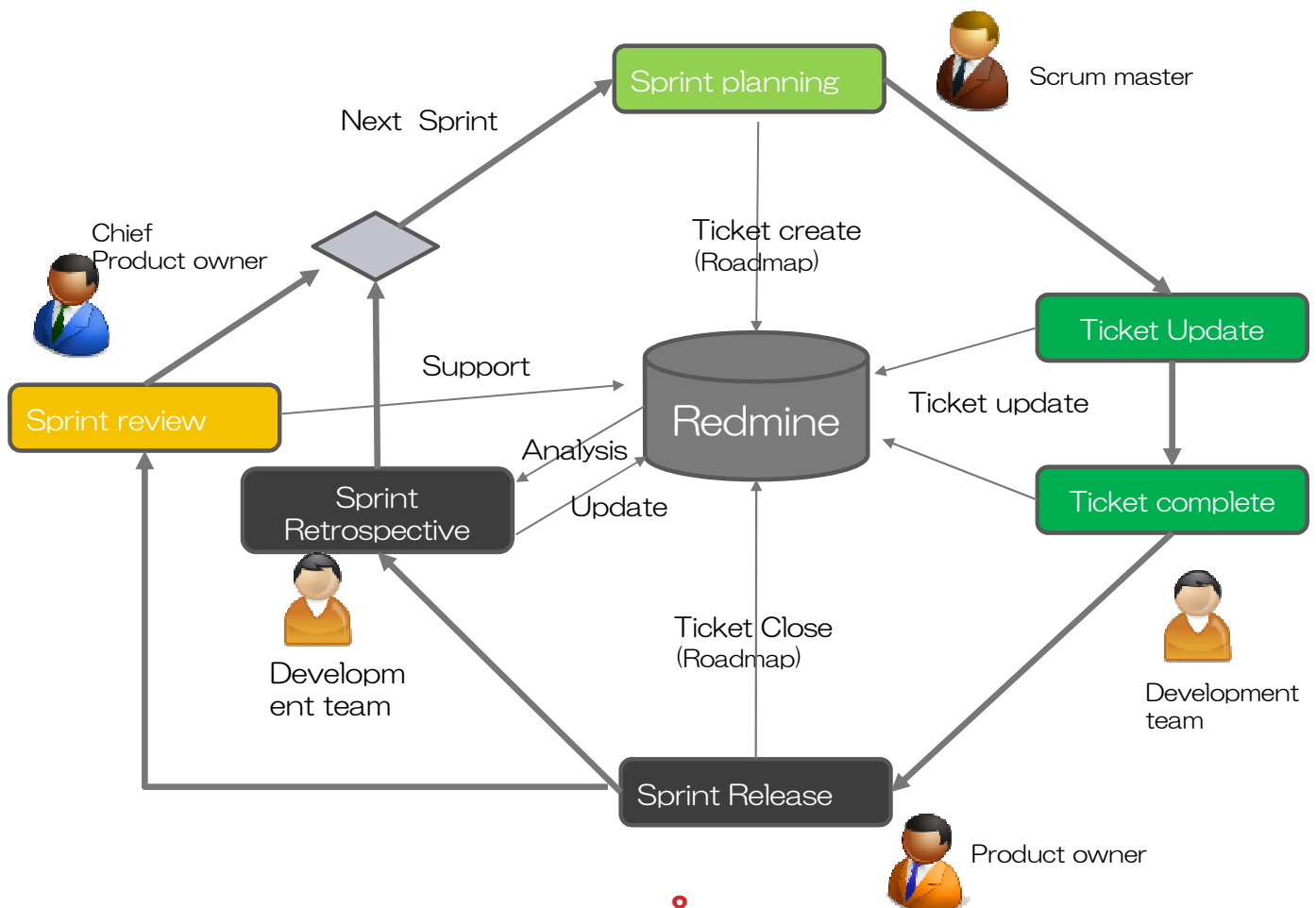
6

5. Scrum framework



7

6. Ticket operative cycle

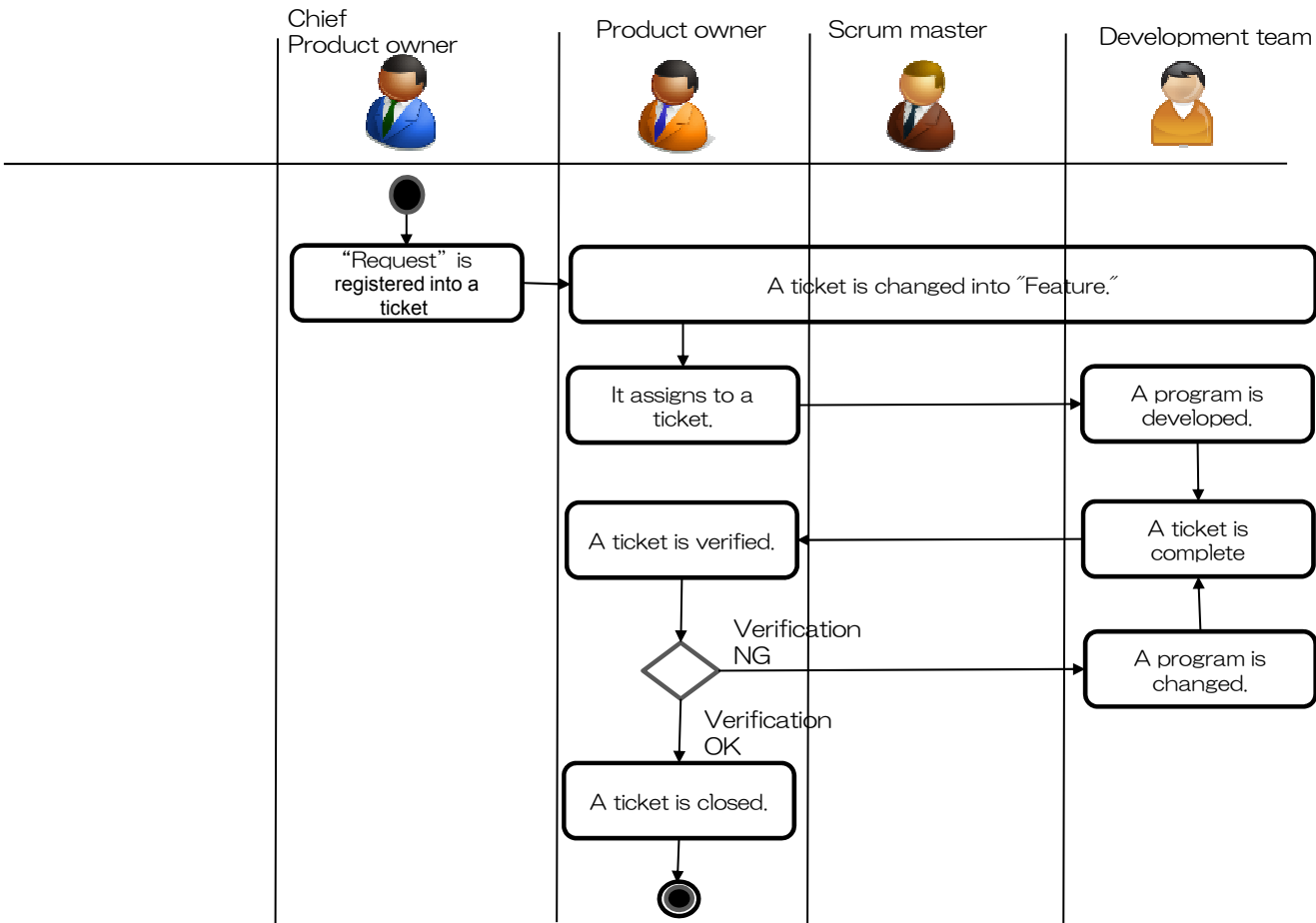


8

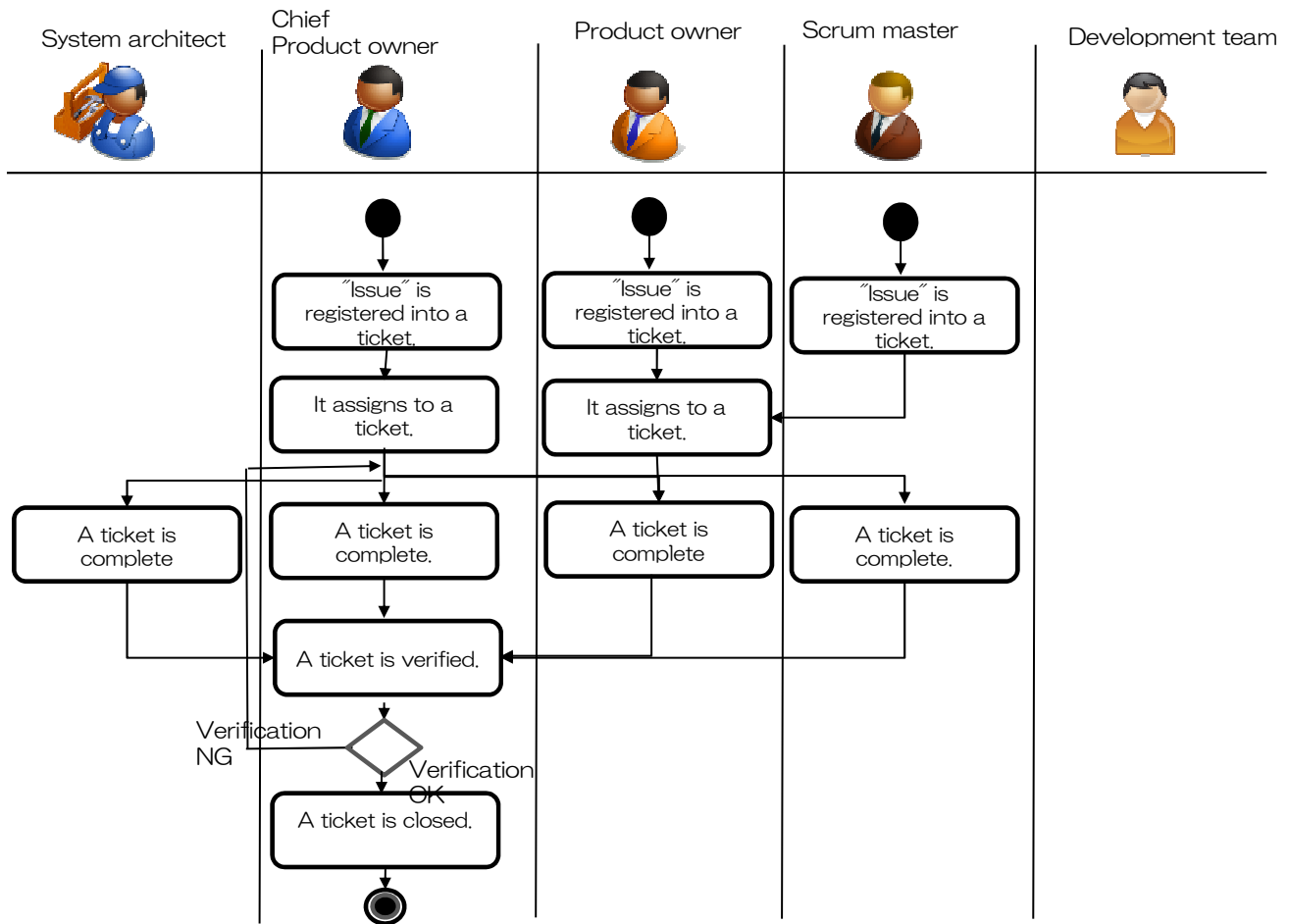
7. Trackers (Ticket kind)

Trackers	Explanation
Request	Ticket of the product backlog User demand. I switch to Feature after adjustment during a man-hour, a period
Feature	Ticket of the sprint backlog Design, programming, ticket of the test
Issue	Problem to have in a development team
Bug	Bug of unit test and Bug of integration test
Support	Question from a user This ticket changes it to Close, Issue, Request, Bug
Note	memo and idea

8.1 Workflow of a tracker (Request • Feature)

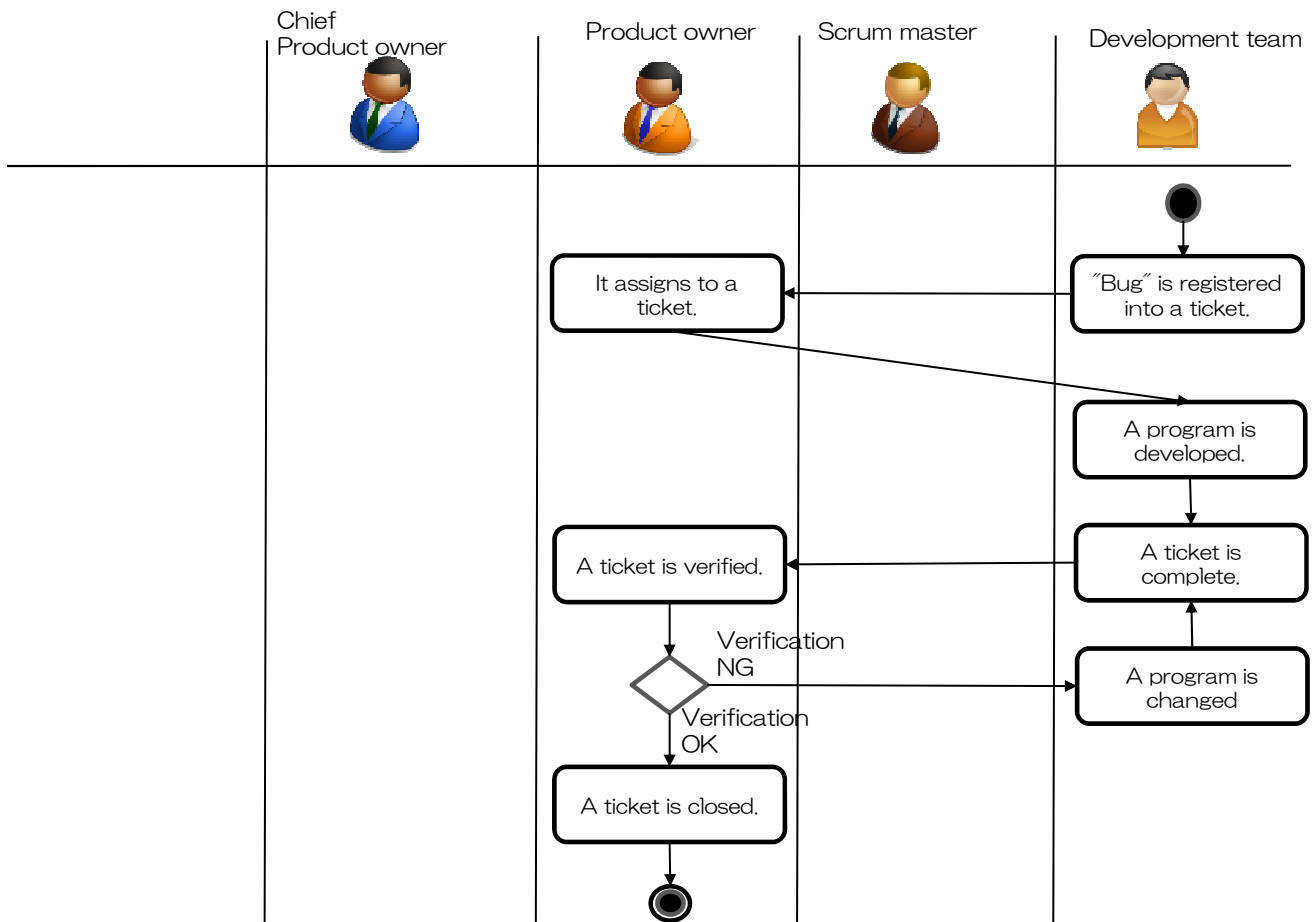


8.2 Workflow of a tracker (Issue)



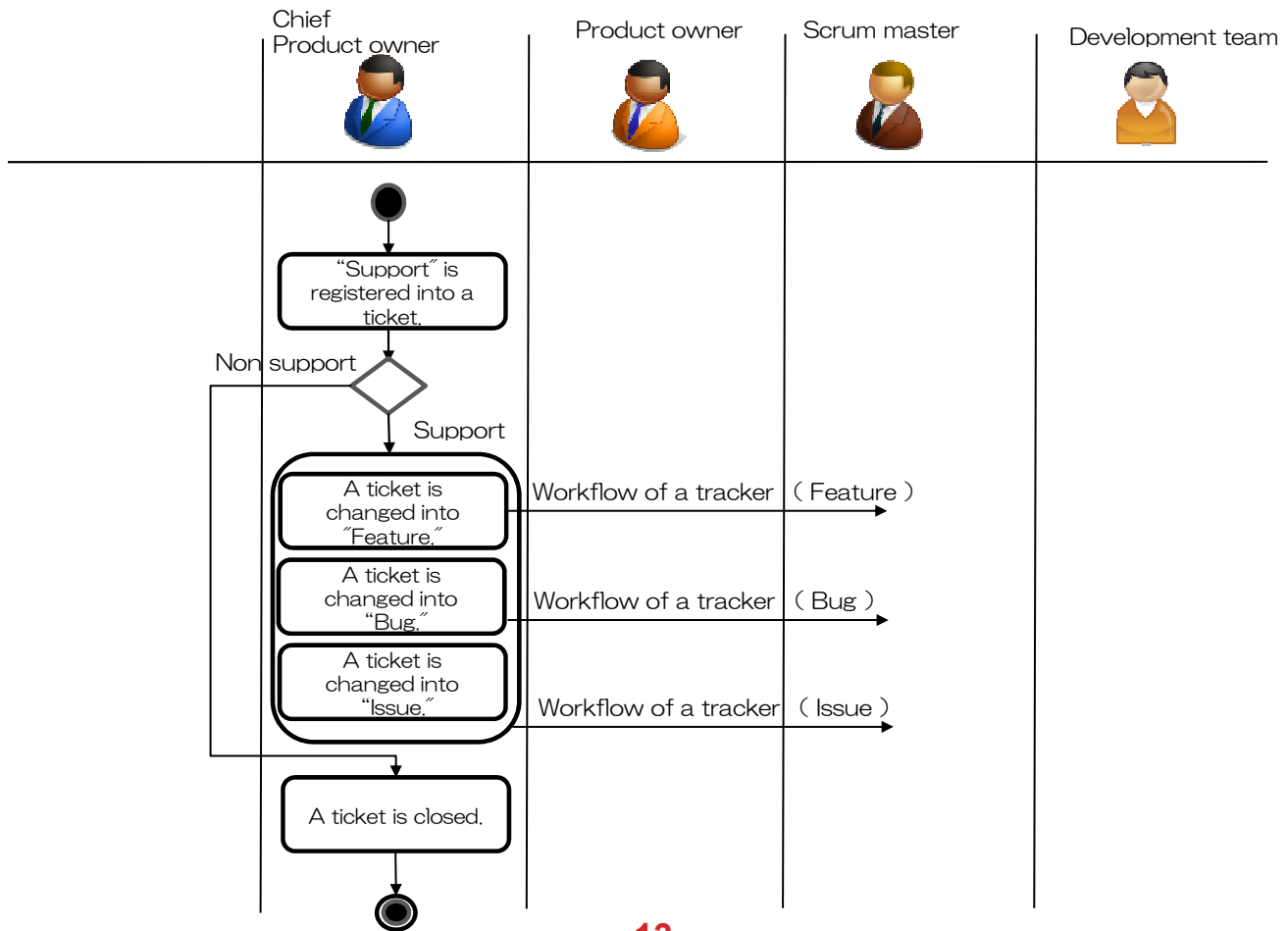
11

8.3 Workflow of a tracker (Bug)



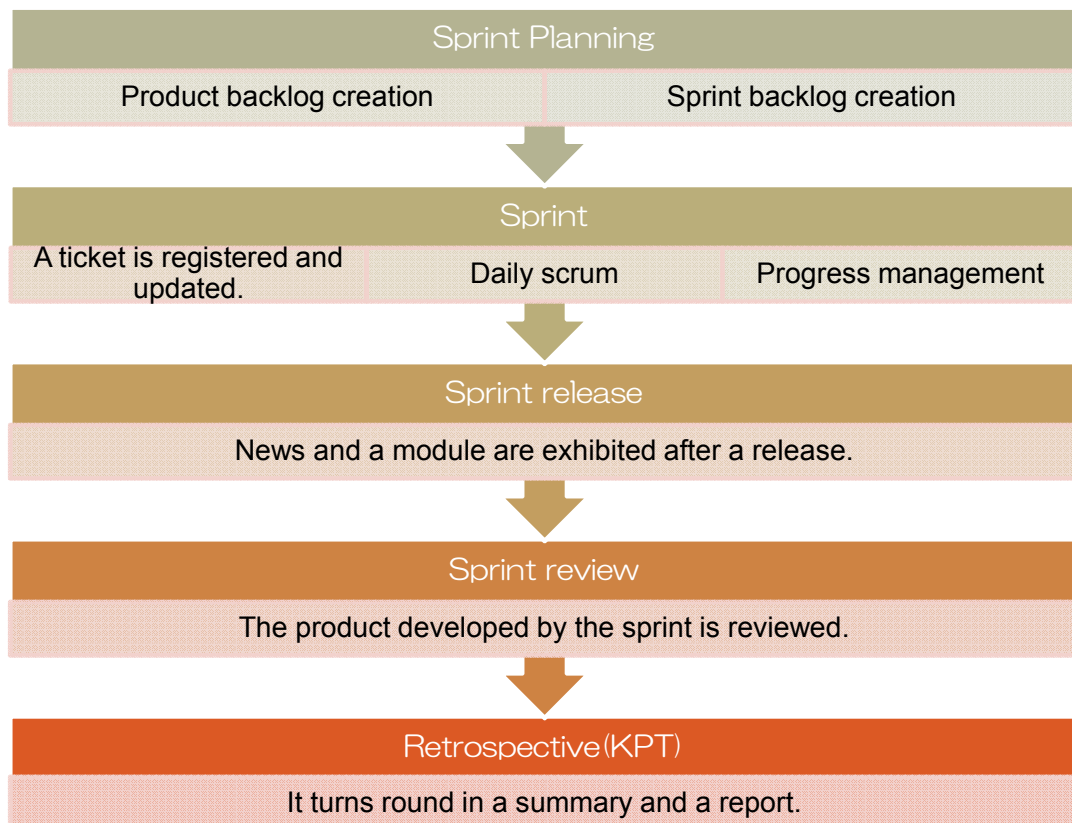
12

8.4 Workflow of a tracker (Support)



13

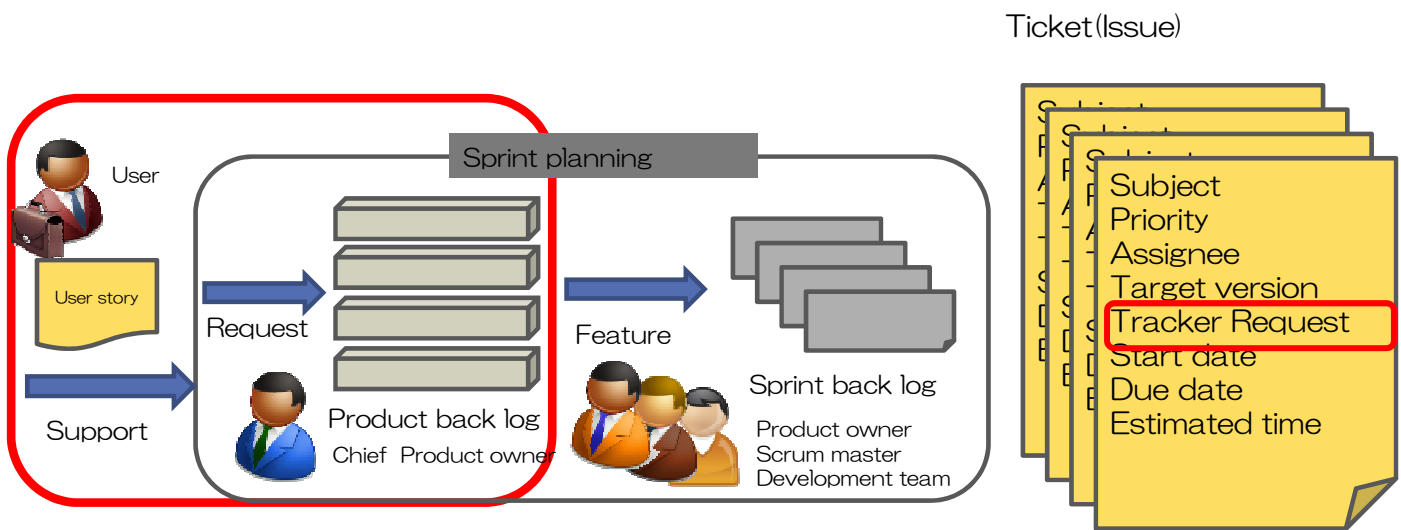
9. Operations Flow of Redmine



14

9.1 Sprint planning(**Product backlog**)

A product backlog is created from a user story.



It sets to a ticket.

- Tracker "Request"

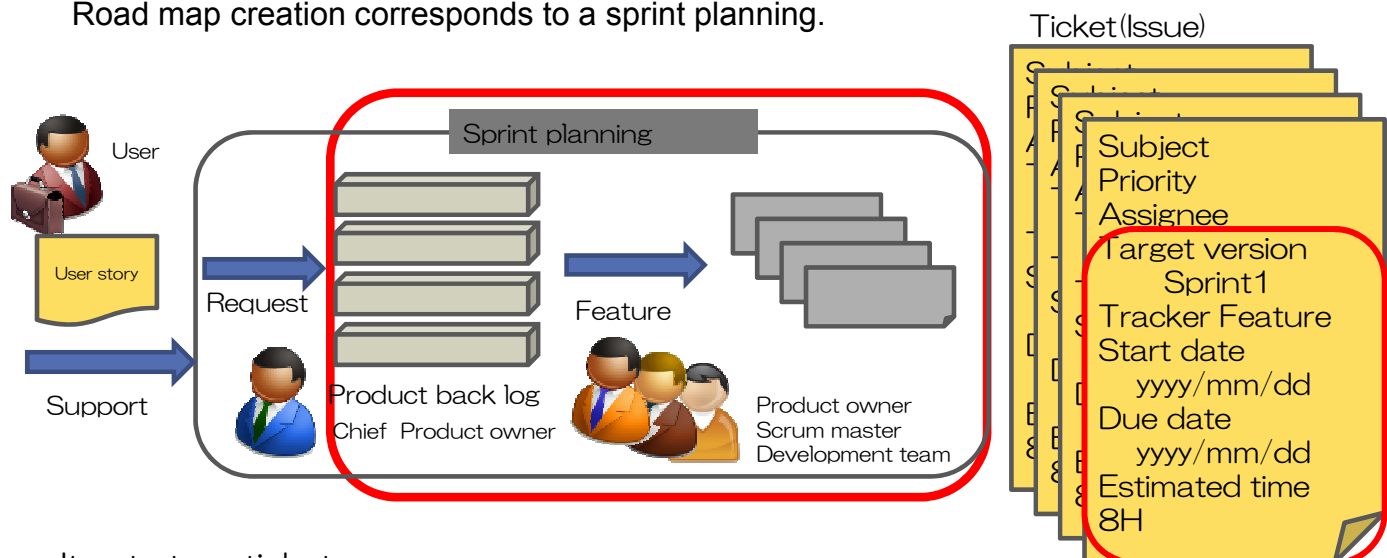
15

9.2 Sprint planning(**Sprint backlog**)

A sprint backlog is created from a product backlog.

It is a sprint backlog = roadmap in Redmine.

Road map creation corresponds to a sprint planning.



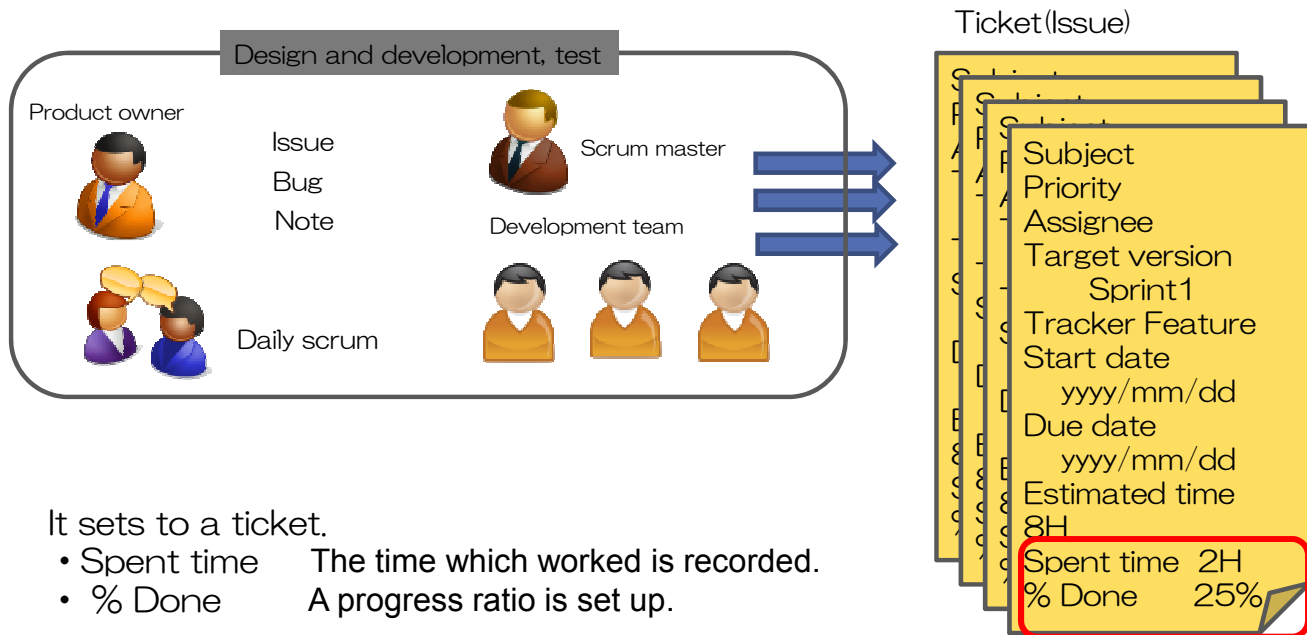
It sets to a ticket.

- Target version Sprint1 or Sprint2
- Tracker Feature
- Start date Start date
- Due date The date of a scheduled end
- Estimated time Estimated time

16

9.3 Sprint (Ticket is registered and updated.)

A ticket is updated in conjunction with development.



It sets to a ticket.

- Spent time The time which worked is recorded.
- % Done A progress ratio is set up.

When a bug occurs, a ticket is created by a tracker "Bug. "

When a Issue occurs, a ticket is created by a tracker "Issue. "

A scrum master sets up Start date , Due date, and Estimated time

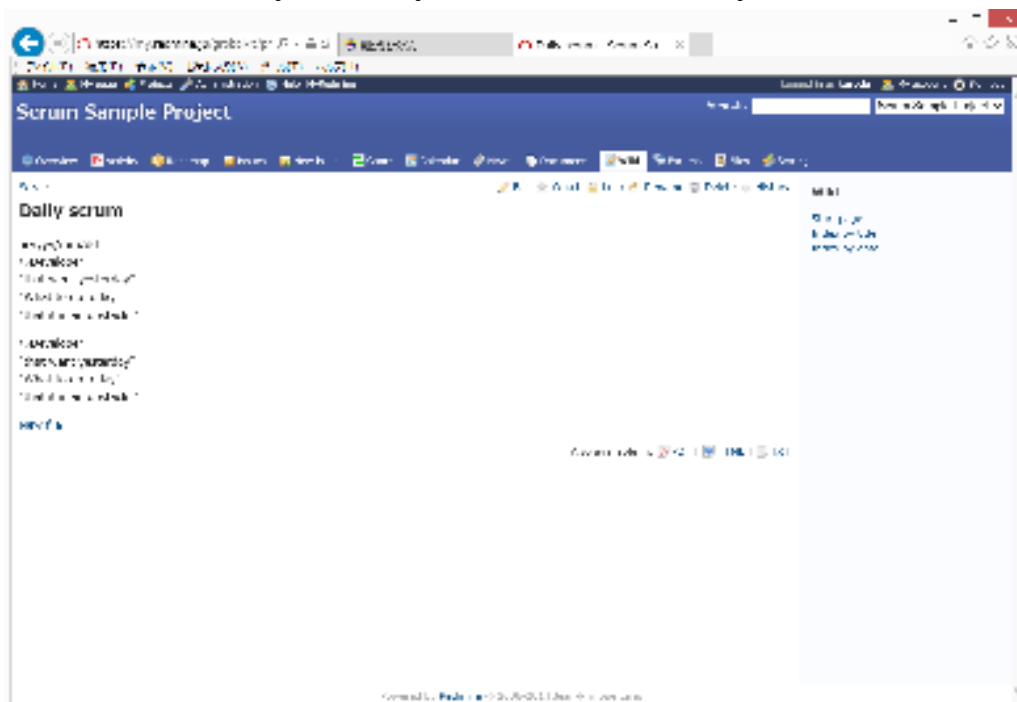
17

9.4 Sprint(Daily scrum)

Daily scrum is performed and a result is opened to Wiki.

It is a meeting which checks progress by all the members.

Share "that went yesterday", "What to do today" and "that it is an obstacle."

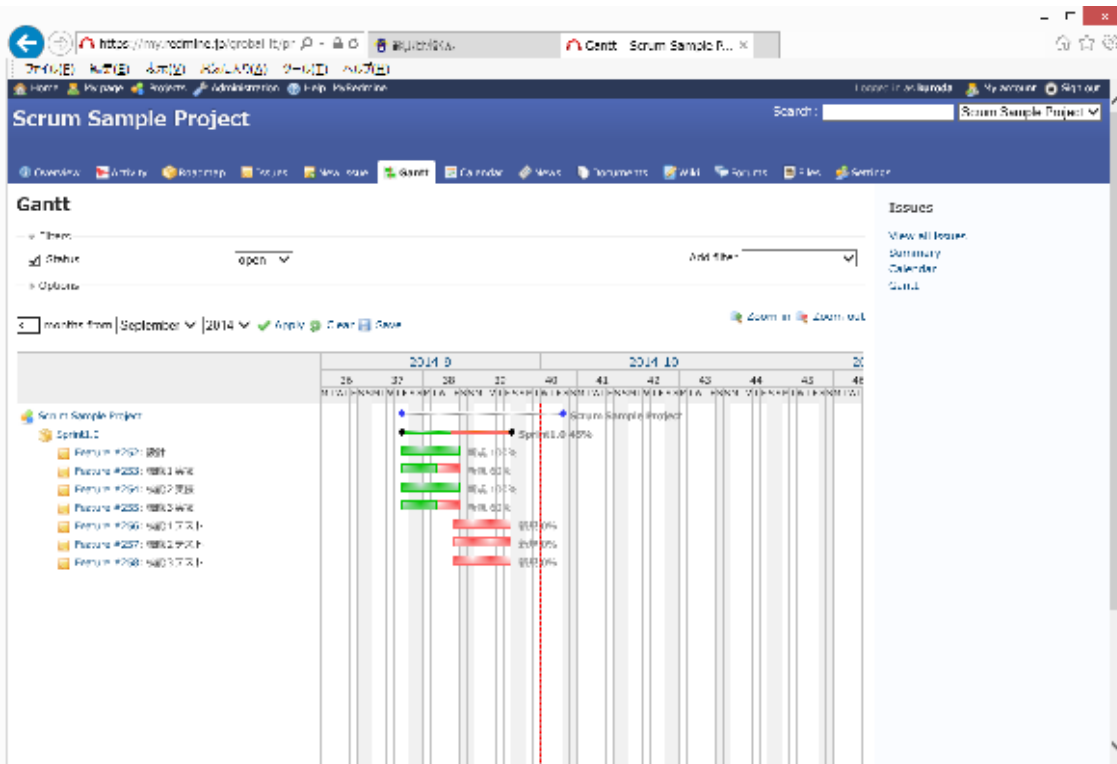


18

9.5 Sprint(Progress management)

Progress is managed by a Gantt chart.

A Gantt chart is created from a schedule man day, working hours, and

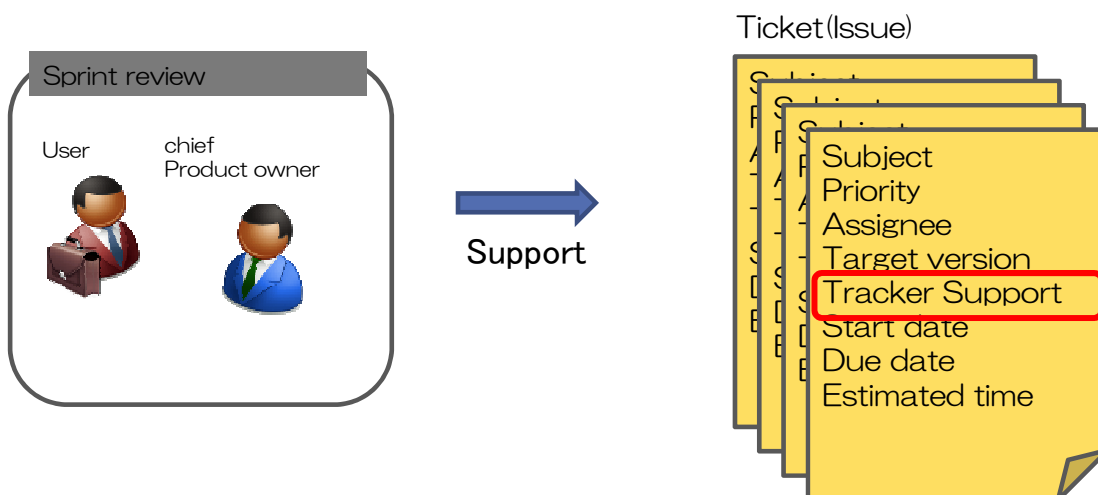


19

9.6 Sprint review

The product developed by the sprint is reviewed.

The product backlog of an improvement request or an obstacle is added if needed.



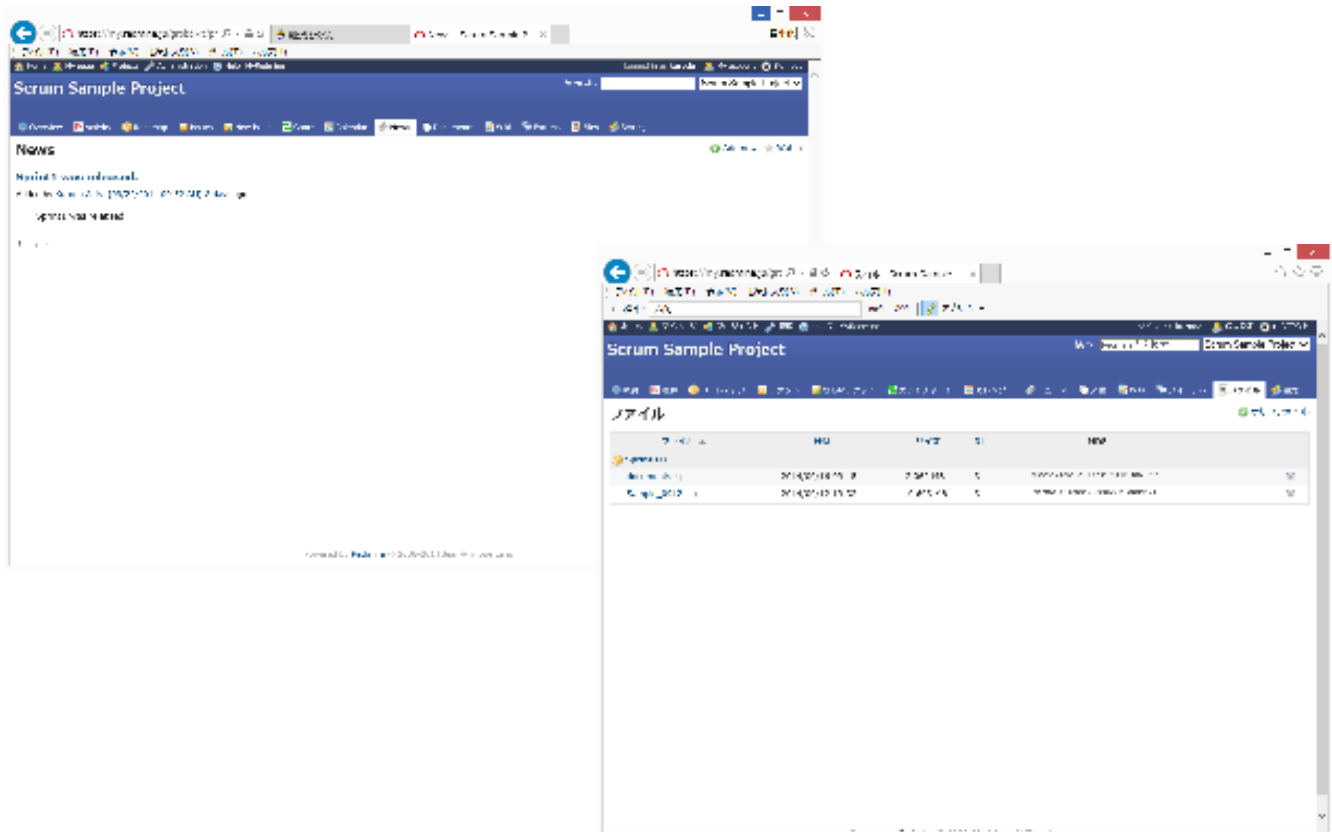
It sets to a ticket.

- Tracker Support

20

9.7 Sprint release

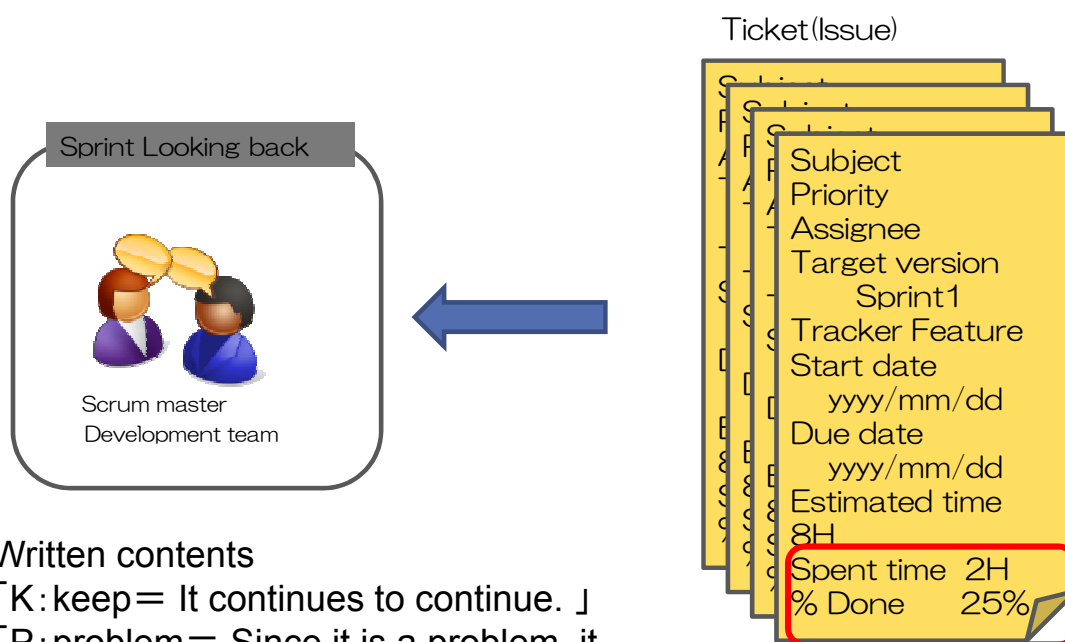
News and a module are exhibited after a release.



21

9.8 Sprint retrospective (KPT)

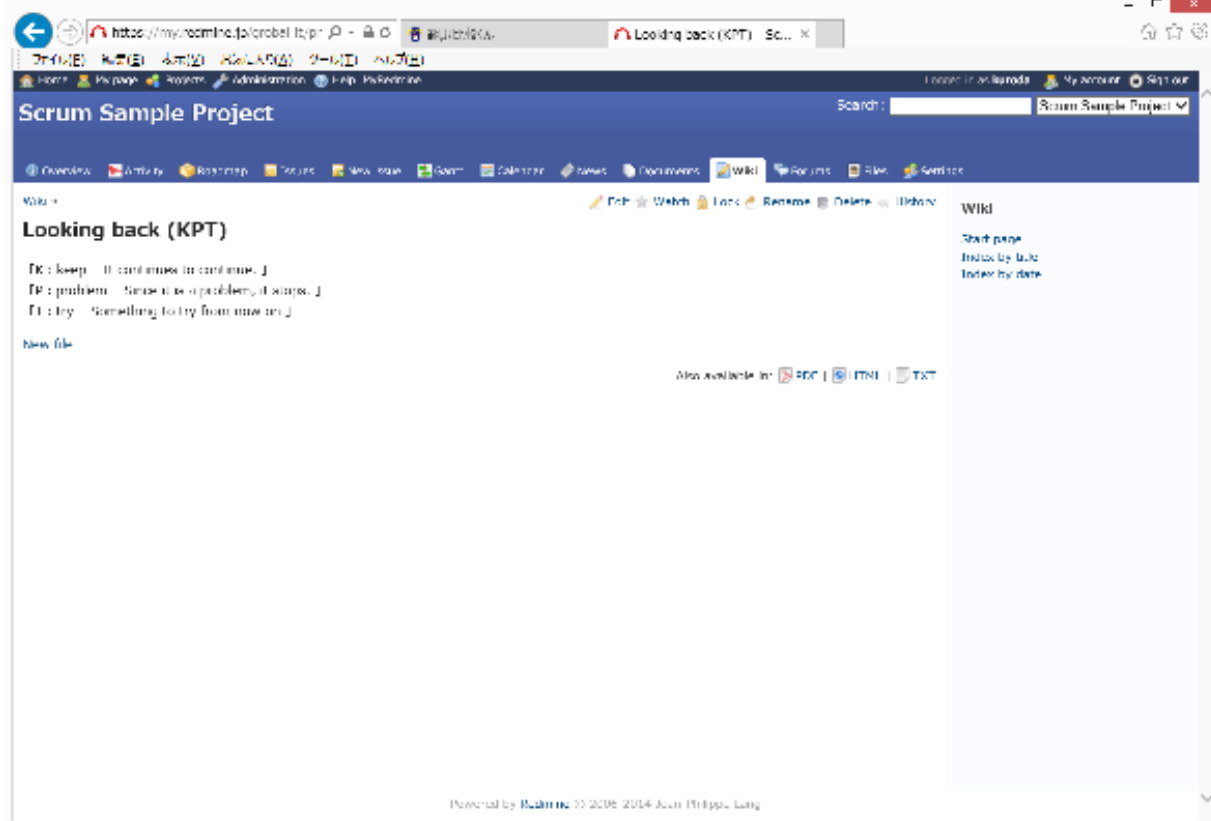
It returns by shaking from a schedule man day and working hours.



22

9.9 Sprint retrospective (KPT) Information sharing

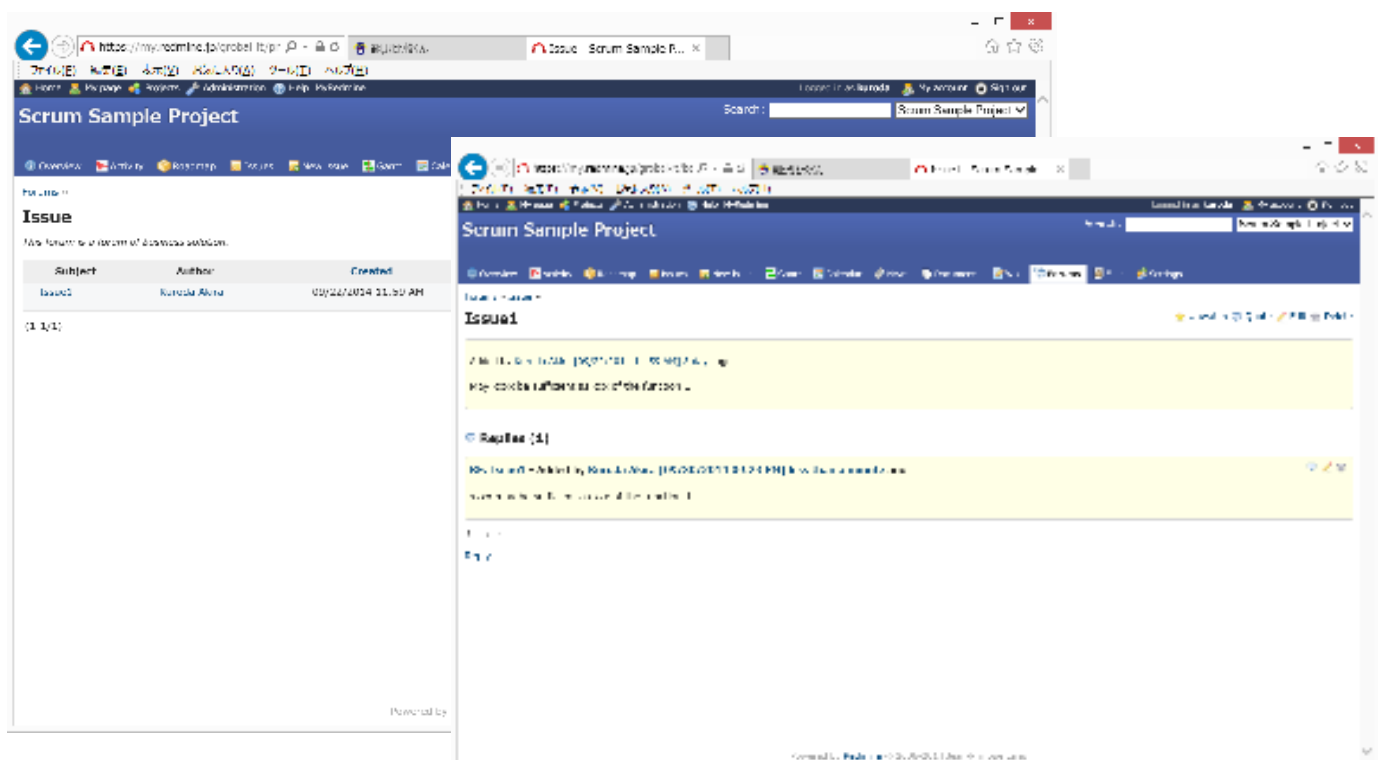
Information sharing of the result upon which it looks back is carried out by Wiki



23

9.10 The solution of a Issue

When an Issue ticket is created, subjects, such as a question of functional specification and a point in question, are argued about them and solved by a forum. An examination result is indicated at an Issue ticket.



24

10.1 Project information for the exercise course

Site URI = <https://~~~~~/>

Project	User	Roll
TeamA	JPN-Usr1	Project manager
	JPN-Usr2	System architect
	JPN-Usr3	Chief Product owner
	OffA-Usr1	Product owner
	OffA-Usr2	Scrum master
	OffA-Usr3	Developer
	OffA-Usr4	Developer
	OffA-Usr5	Developer
TeamB	JPN-Usr1	Project manager
	JPN-Usr2	System architect
	JPN-Usr4	Chief Product owner
	OffB-Usr1	Product owner
	OffB-Usr2	Scrum master
	OffB-Usr3	Developer
	OffB-Usr4	Developer
	OffB-Usr5	Developer

※Modify for your environment.

25

10.2 User information for the exercise course

User	Password	Display name	Mail address	Language
JPN-Usr1	password	JPN Usr1	JPN-Usr1@xxxx.co.jp	Japanese
JPN-Usr2	password	JPN Usr2	JPN-Usr2@xxx.co.jp	Japanese
JPN-Usr3	password	JPN Usr3	JPN-Usr3@xxx.co.jp	Japanese
JPN-Usr4	password	JPN Usr4	JPN-Usr4@xxx.co.jp	Japanese
OffA-Usr1	password	OffA Usr1	PHI-Usr1@xxx.co.jp	English
OffA-Usr2	password	OffAI Usr2	PHI-Usr2@xxx.co.jp	English
OffA-Usr3	password	OffA Usr3	PHI-Usr3@xxx.co.jp	English
OffA-Usr4	password	OffA Usr4	PHI-Usr4@xxx.co.jp	English
OffA-Usr5	password	OffA Usr5	PHI-Usr5@xxx.co.jp	English
OffB-Usr1	password	OffB Usr1	IHA-Usr1@xxx.co.jp	English
OffB-Usr2	password	OffB Usr2	IHA-Usr2@xxx.co.jp	English
OffB-Usr3	password	OffB Usr3	IHA-Usr3@xxx.co.jp	English
OffB-Usr4	password	OffB Usr4	IHA-Usr4@xxx.co.jp	English
OffB-Usr5	password	OffB Usr5	IHA-Usr4@xxx.co.jp	English

※Please change a password at the time of first time login.

※Modify for your environment.

26

10.3 User information modification method

When you change a display name, a mail address, and a language, please change into Redmine on the screen of the login back and the following.

The screenshot shows the Redmine web interface. The browser address bar displays `https://my.redmine.jp/label/10m`. The page title is "Redmine". The user is logged in as "PHD-User1". The "My account" page is displayed, with the "Information" tab selected. A red box highlights the following fields:

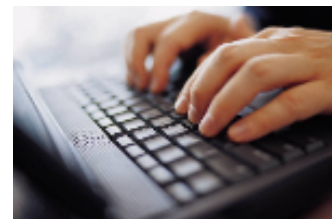
- First name: [First name]
- Last name: [Last name]
- Email: [Email address]
- Language: [English]

Other visible fields and options include:

- Change password
- My account
- Logout
- Search
- Jump to a project...
- Information
- First name
- Last name
- Email
- Language
- Send
- Email notifications
- Only for things I watch or I'm involved in
- I don't want to be notified of changes that I make myself
- Preferences
- Hide my email address
- Time zone
- Display comments in chronological order
- Warn me when leaving a page with unsaved text
- My account
- Logout
- Created: 03/26/10 14:03:13 PM
- Delete my account
- Atom access key
- Atom access key created about 2 hours ago (Watch)
- API access key
- Show
- API access key created less than a minute ago (Watch)



地方型グローバルIT人材の育成 PBLカリキュラム Redminの使用方法



富山情報ビジネス専門学校編

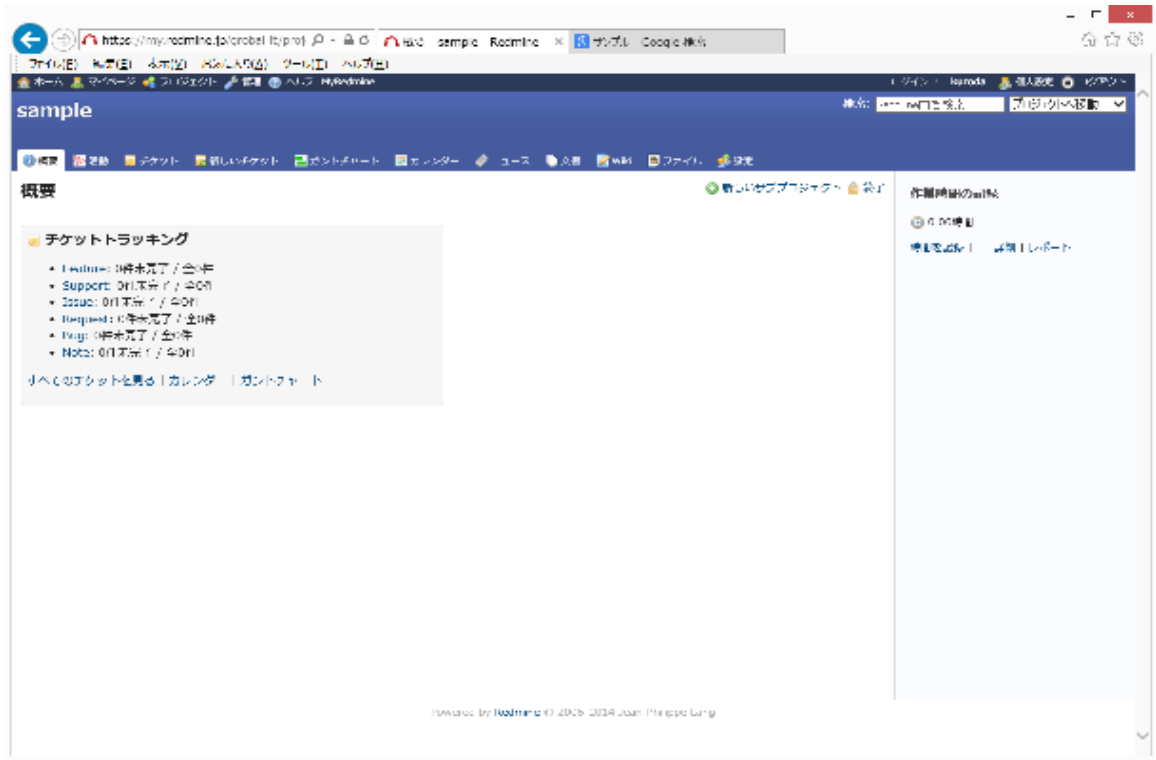


目次

1.Redmineの概要	3
2.Redmineの機能概要	4
3.プロジェクト体制図	5
4.プロジェクトにおける役割	6
5.Scrumフレームワーク	7
6.チケット運用サイクル	8
7.トラッカー（チケットの種類）	9
8.トラッカーのワークフロー	
8.1.トラッカーのワークフロー（Request・Feature）	10
8.2.トラッカーのワークフロー（Issue）	11
8.3.トラッカーのワークフロー（Bug）	12
8.4.トラッカーのワークフロー（Support）	13
9.運用フロー	14
9.1.スプリントプランニング（プロダクトバックログ）	15
9.2.スプリントプランニング（スプリントバックログ）	16
9.3.スプリント（チケットを登録、更新）	17
9.4.スプリント（デイリースクラム）	18
9.5.スプリント（進捗管理）	19
9.6.スプリントレビュー	20
9.7.スプリントリリース	21
9.8.振り返り(KPT)	22
9.9.振り返り(KPT) 情報共有	23
9.10.課題の解決方法	24
10.プロジェクト情報	
10.1.プロジェクト情報	25
10.2.ユーザ情報	26
10.3.ユーザ情報変更方法	27

1.Redmineの概要

Webベースでプロジェクトで実施すべきタスクの管理、情報の集約管理ができるプロジェクト管理システムです。
本資料では、RedmineでSurum開発を行うための運用方法について説明します。



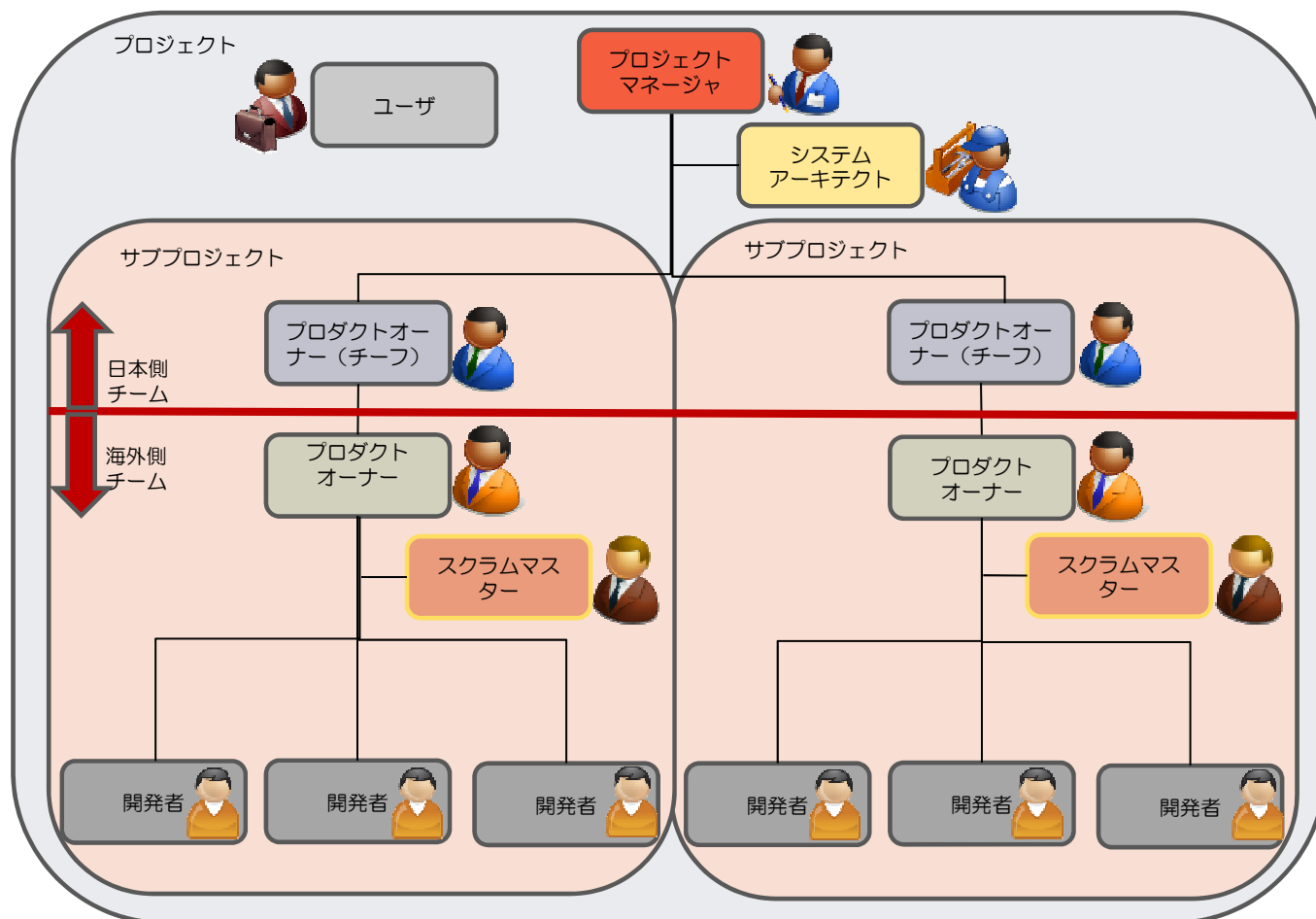
3

2.Redmineの機能概要

機能	概要
概要	概要画面は、プロジェクト全体の情報の要約が表示されます。
活動	活動画面は、プロジェクトにおける全ての作業の時系列による記録を表示されます。
ロードマップ	ロードマップ画面は、プロジェクトのスプリントで区切られたチケットの一覧、及び進捗が表示されます。
チケット	チケット管理はRedmineの中核をなす機能です。 プロジェクトで実施すべき業務を小さなタスクに分解し、Redmine上で「チケット」として登録します。
ガントチャート	ガントチャートには開始日と期日が設定されているスプリントに関連づけられているチケットが表示されます。
カレンダー	チケットとスプリントがカレンダー上に表示されます。
ニュース	プロジェクトメンバーへの伝達事項を掲載できる機能です。 今回は、プロダクトリリース時のアナウンス等に使用します。
文書	プロジェクト内で共有すべきファイルを掲載する機能です。 今回は、構築手順などを格納するために使用します。
Wiki	情報掲載用のページを手軽に追加、更新できる機能です。 今回は、議事録などのメモをチーム内で共有するために使用します。
フォーラム	プロジェクト内で特定の話題について議論を行う機能です。 今回は、課題解決のために議論する場として使用します。
ファイル	リリースしたファイルなどを格納しておき、チーム内でダウンロードできます。 今回は、リリースするプロダクトを格納するために使用します。

4

3.プロジェクト体制図



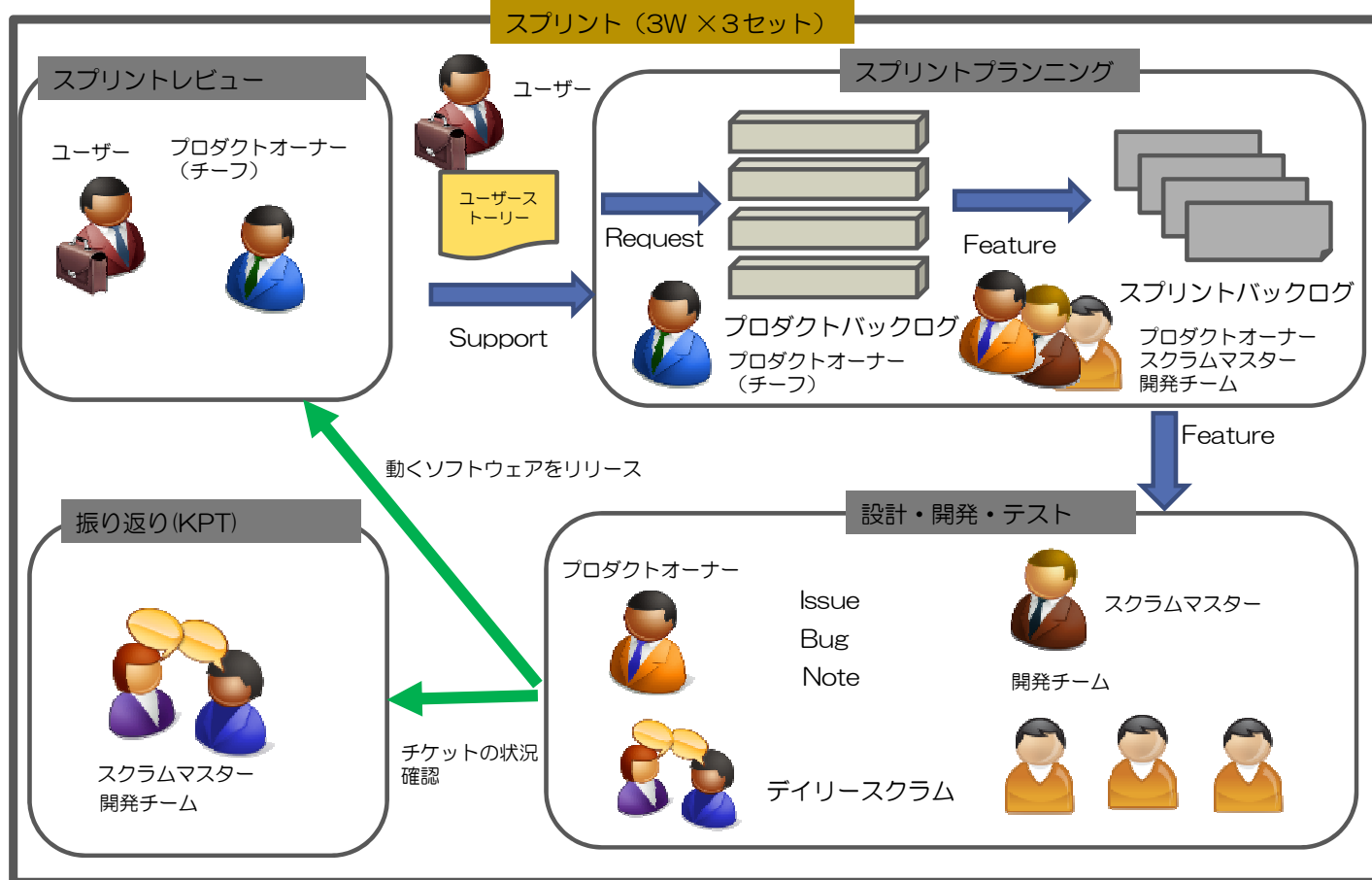
5

4.プロジェクトにおける役割

役割	日本	海外	概要
プロジェクトマネージャ 	○		プロジェクト全体を統括する人物
システムアーキテクト 	○		技術的な問題点や課題の対応をする人物 プロダクトオーナー（チーフ）のサポート役
ユーザー 	○		開発するアプリケーションのユースケース（要望事項）を提示する人物。
プロダクトオーナー（チーフ） 	○		開発するアプリケーションの仕様を決定する人物で、機能仕様上で問題や疑問点が発生した場合に、機能仕様を最終的に判断して決定する。
プロダクトオーナー 		○	開発するアプリケーションの機能仕様上で問題や疑問点が発生した場合に、プロダクトオーナー（チーフ）に問い合わせを行う人物でスプリントリリースの判定も行う。
スクラムマスター 		○	スプリントバックログが更新されているかチェックする人物で、チームの効率が最大になるように配慮し、問題が発生した場合の相談役としての役割も持つ。
開発チーム 		○	スプリントバックログの更新を行う人物で、実装とテストの能力を持つ。

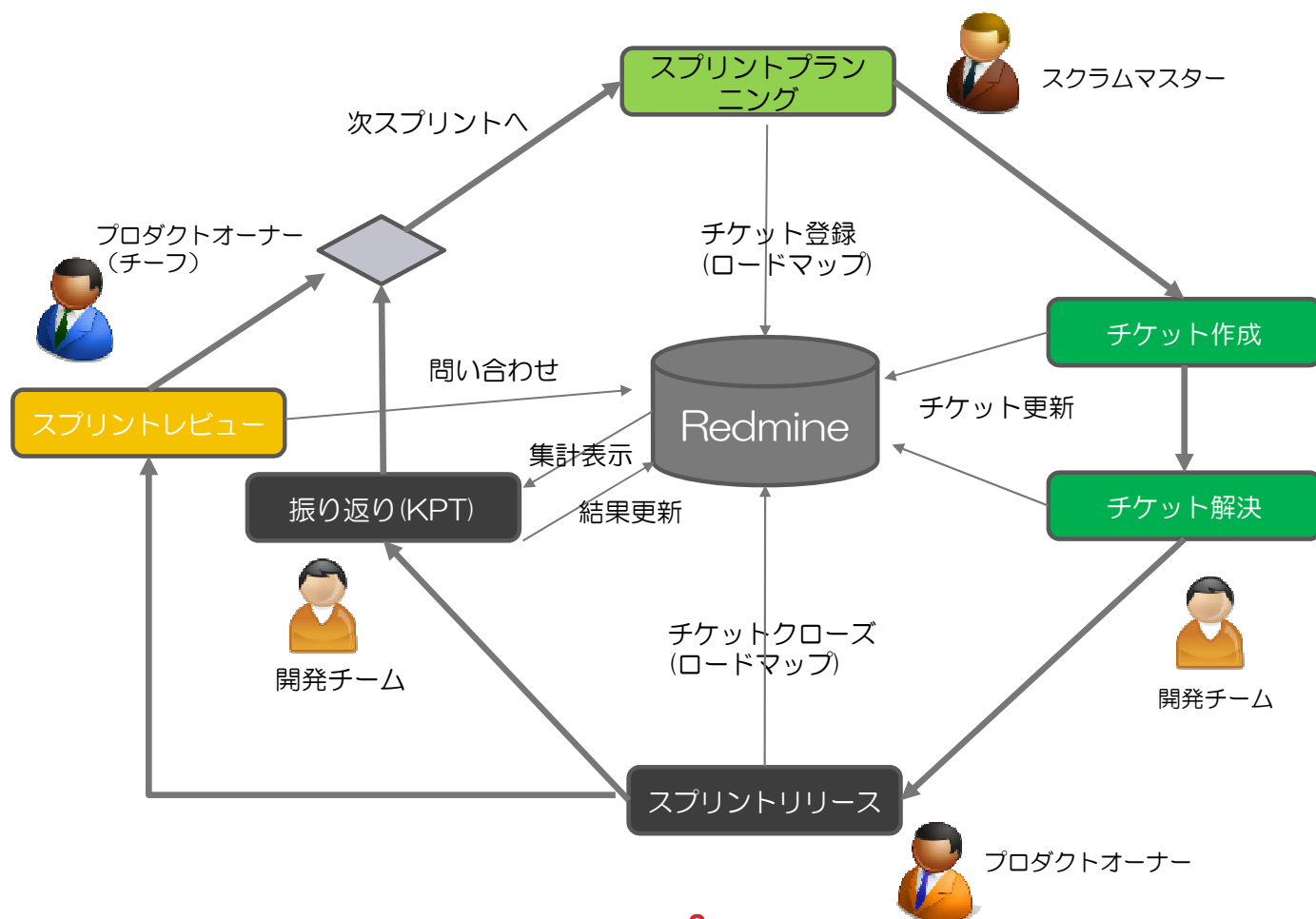
6

5.Scrumフレームワーク



7

6.チケット運用サイクル



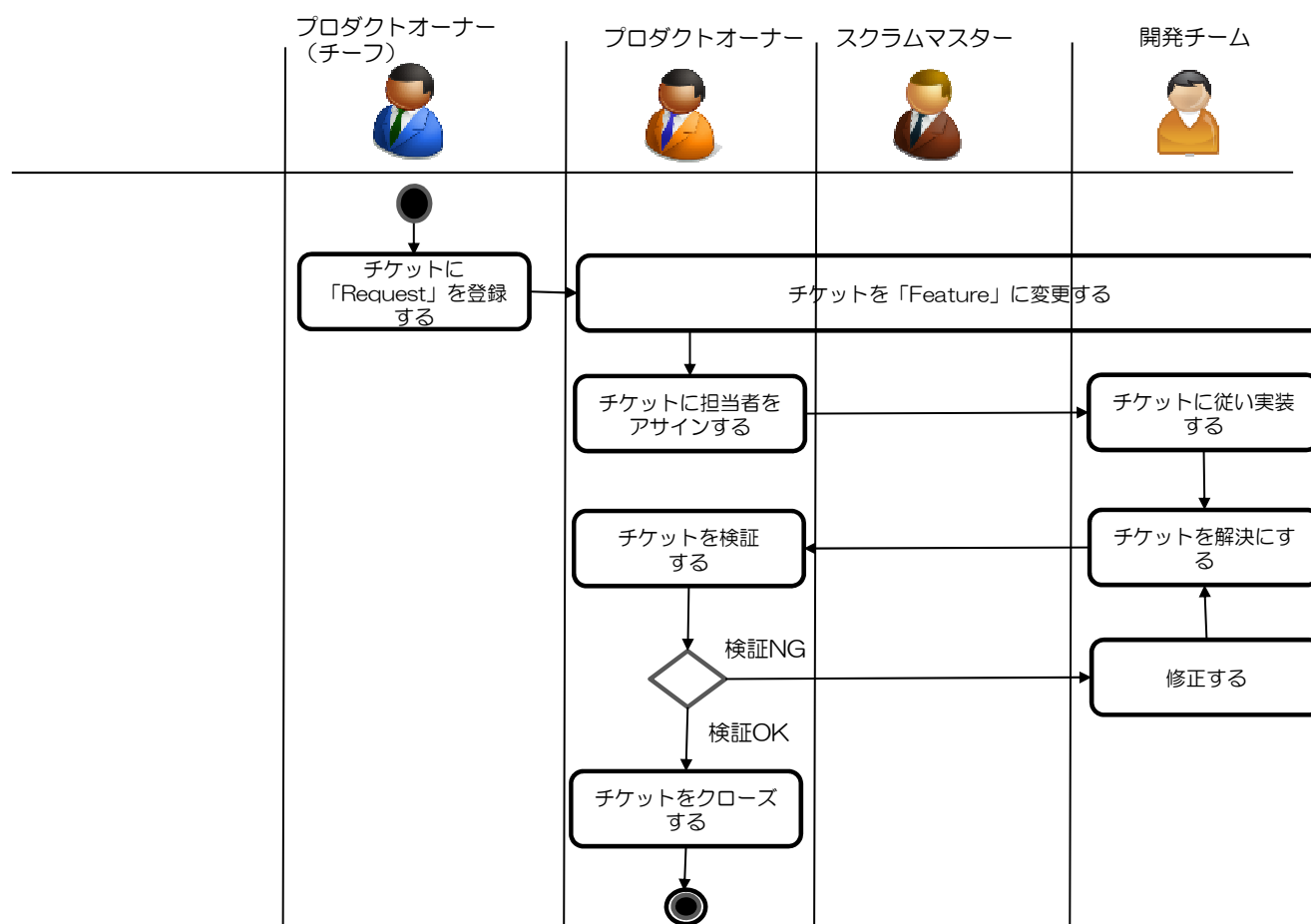
8

7.トラッカー(チケットの種類)

トラッカー	説明
Request	プロダクトバックログのチケット ユーザー要望など。このチケットにより要件が管理され、実現可否・工数・期間などの調整後に Feature に変更していく。
Feature	スプリントバックログのチケット 設計やプログラミング、テストなど
Issue	開発チームで抱える課題。
Bug	障害全般。 単体テスト障害、結合テスト障害などの障害
Support	ユーザーからの問い合わせなど。このチケットは Close するか、または IssueやRequest や Bug に変更していく。
Note	メモやアイデア。

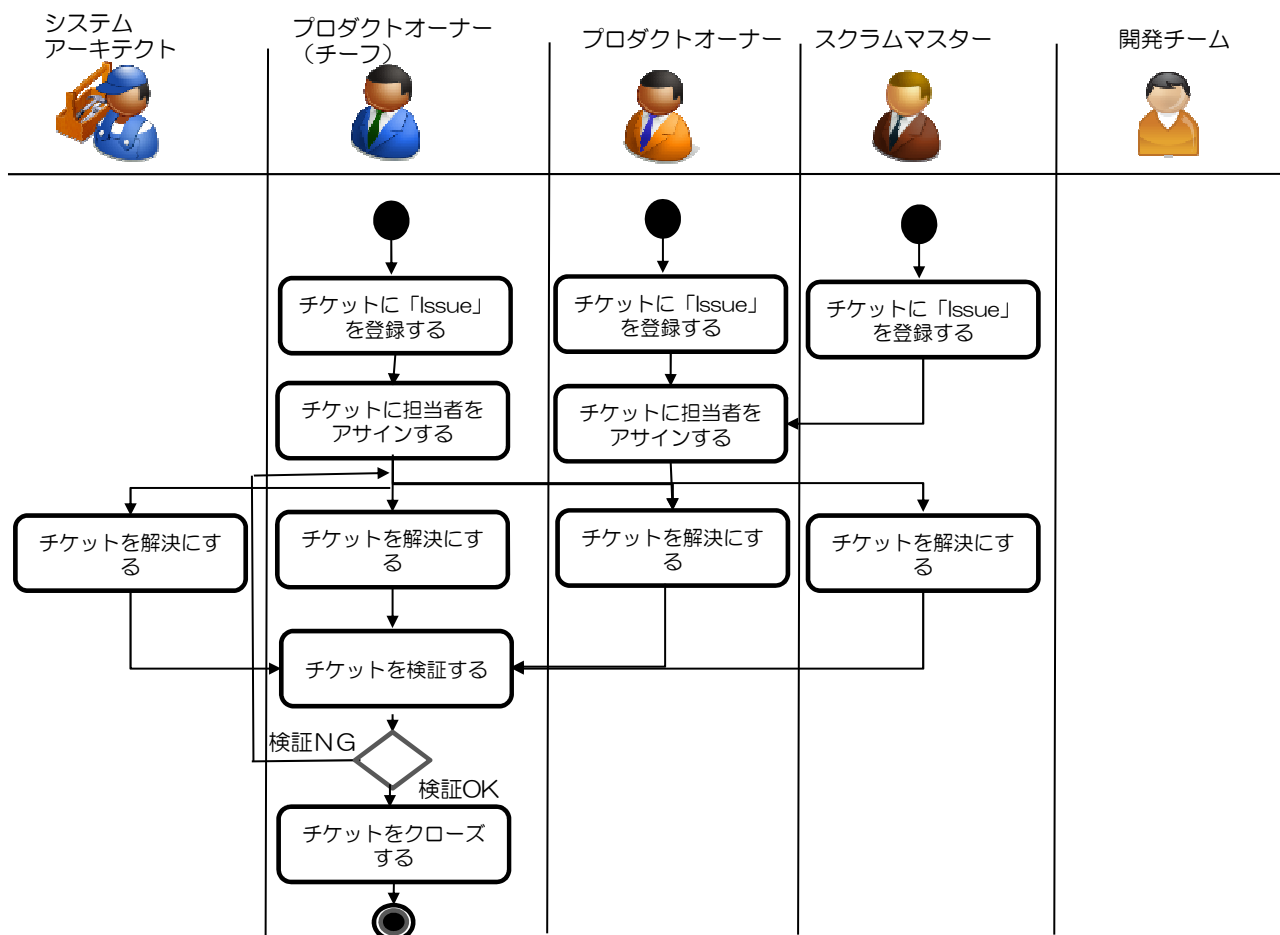
9

8.1 トラッカーのワークフロー (Request・Feature)



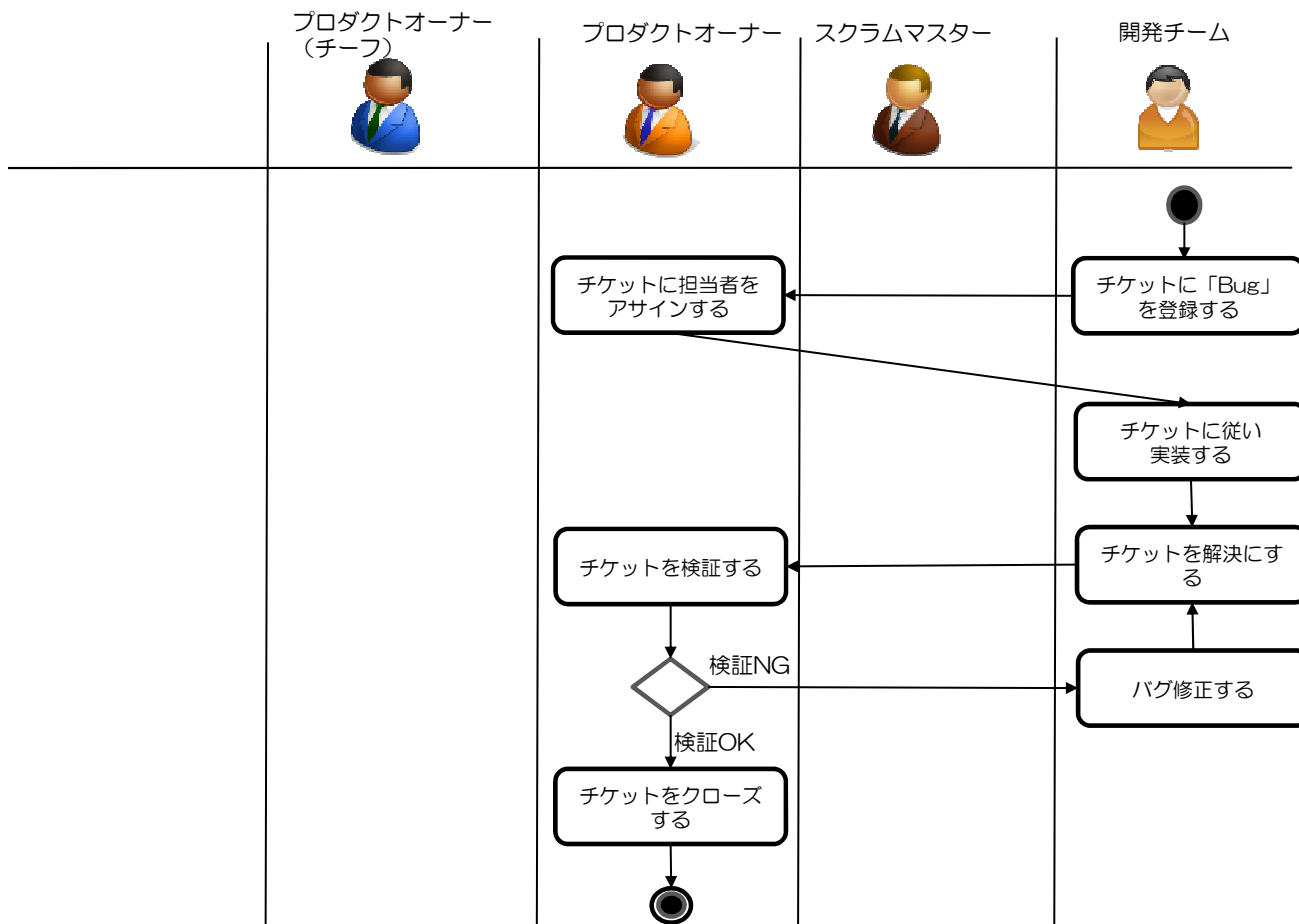
10

8.2 トラッカーのワークフロー (Issue)



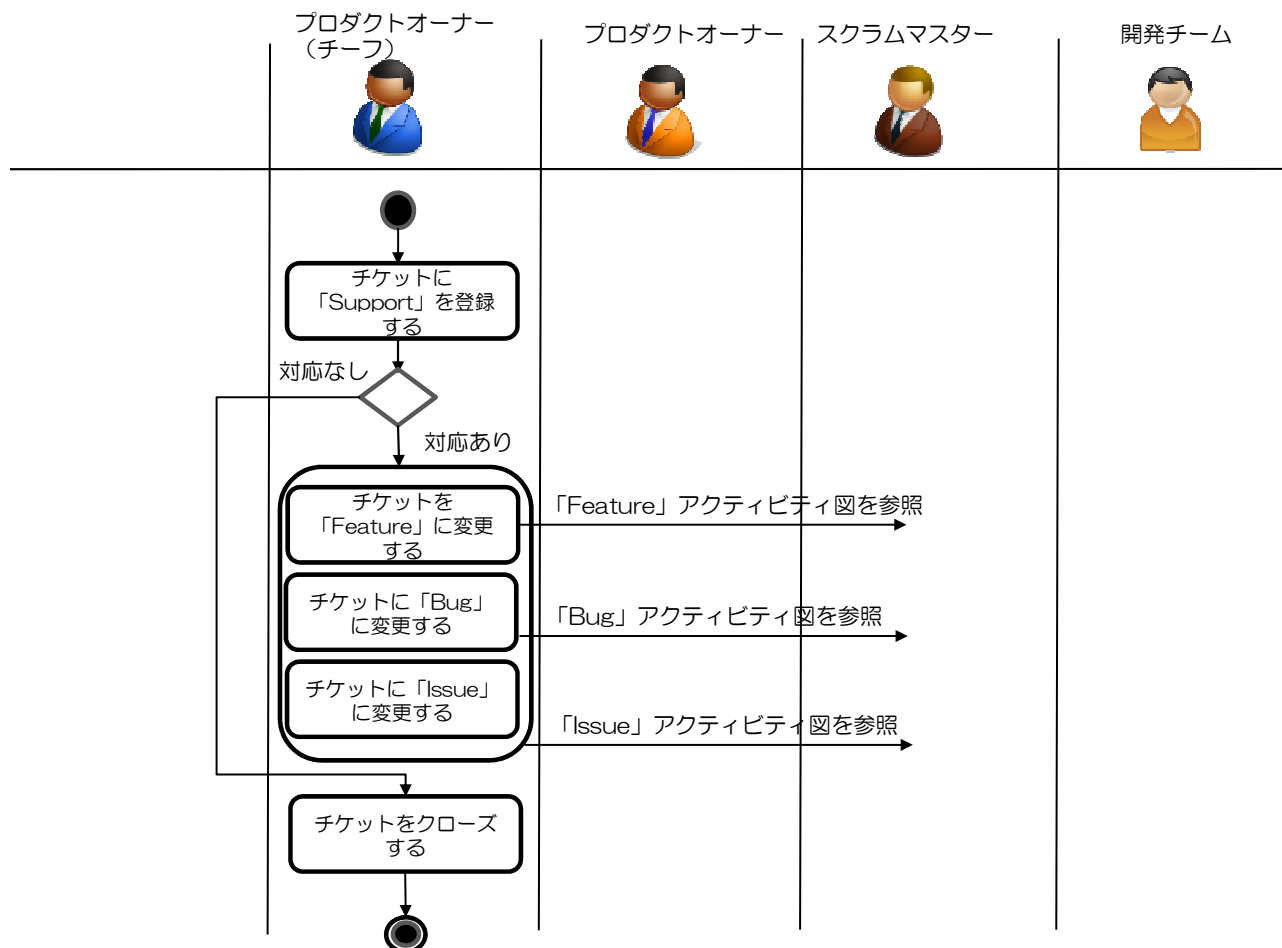
11

8.3 トラッカーのワークフロー (Bug)



12

8.4 トラッカーのワークフロー (Support)



13

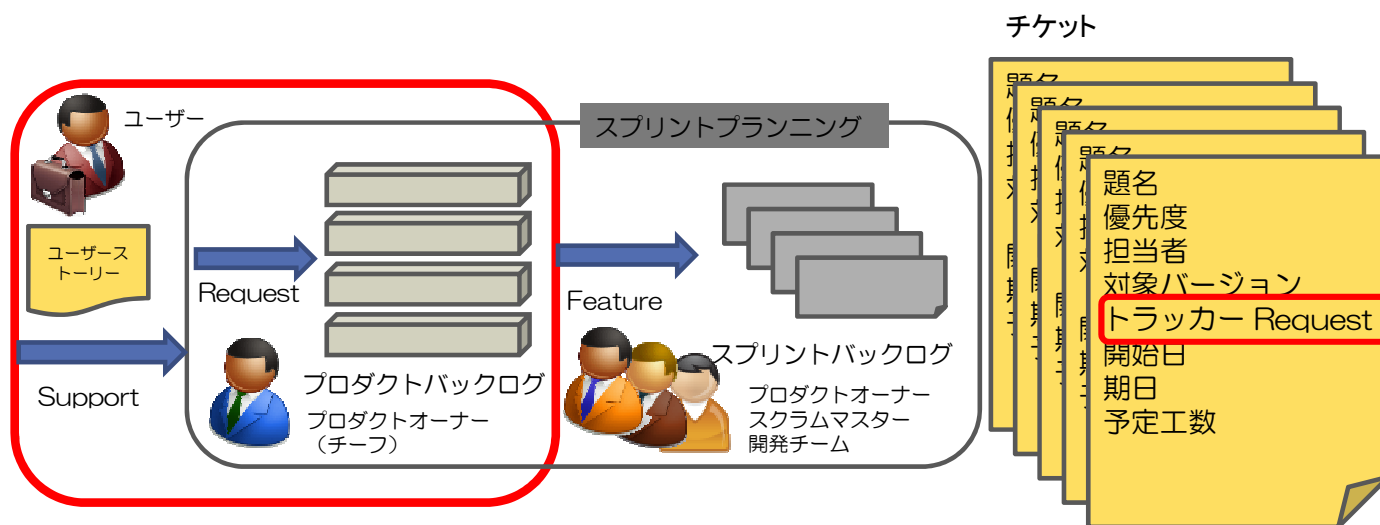
9.Redmineの運用フロー



14

9.1 スプリントプランニング(プロダクトバックログ)

ユーザーストーリーからプロダクトバックログを作成します。



チケット項目に以下の内容を設定します。

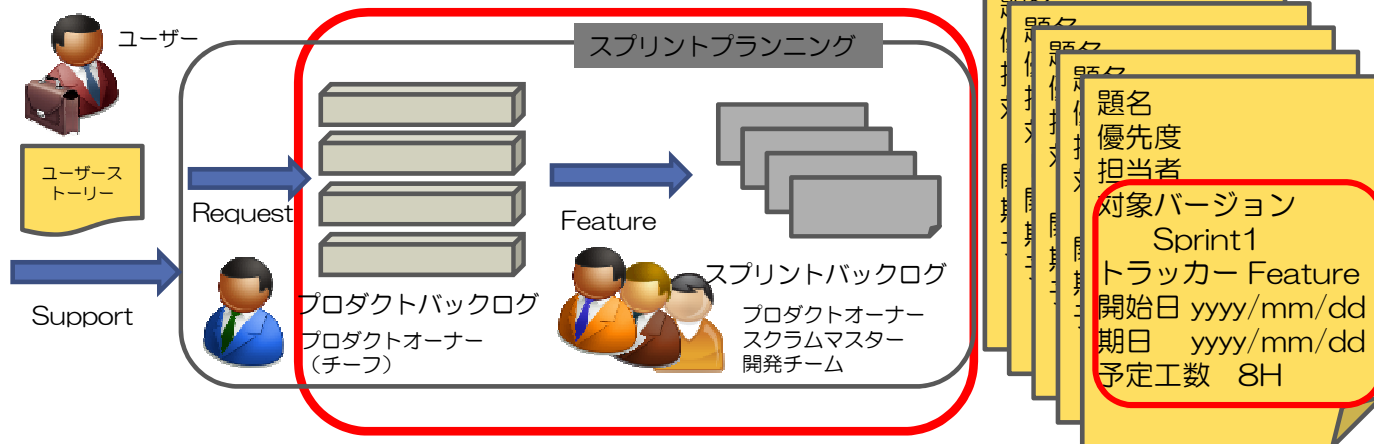
- ・トラッカー Request

15

9.2 スプリントプランニング(スプリントバックログ)

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します。

Redmineではスプリントバックログ=ロードマップです。
ロードマップ作成がスプリントプランニングに該当します。



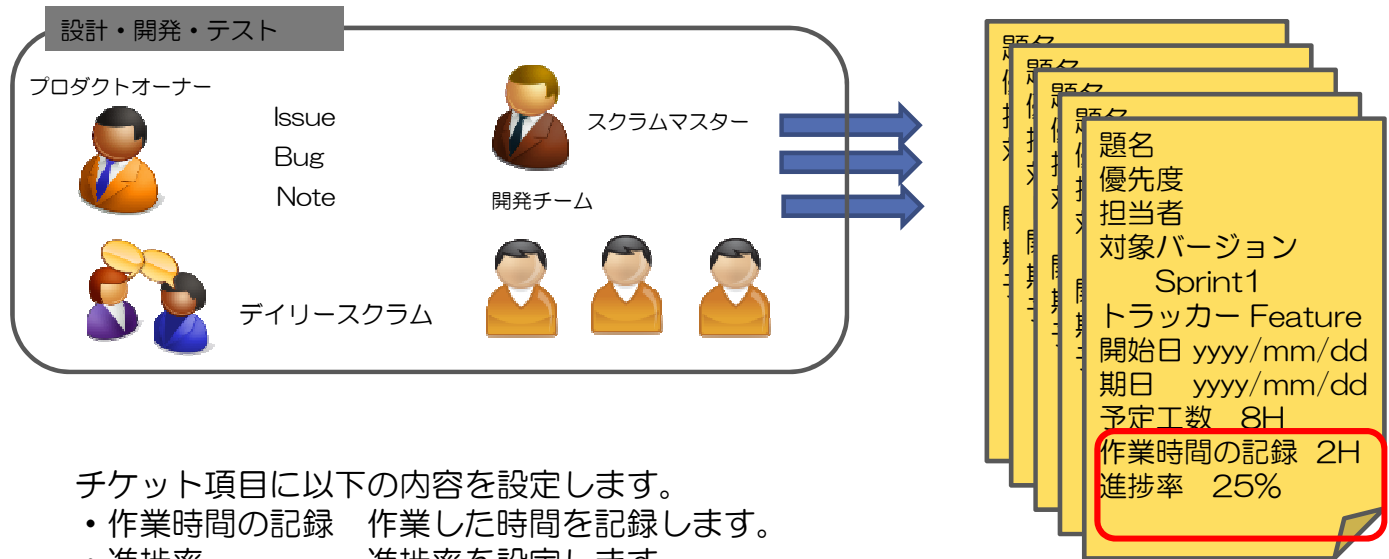
チケット項目に以下の内容を設定します。

- ・対象バージョン Sprint1またはSprint2
- ・トラッカー Feature
- ・開始日 開始日
- ・期日 終了予定日
- ・予定工数 予定工数

16

9.3 スプリント(チケットを登録、更新)

開発と連動してチケットを更新します。



チケット項目に以下の内容を設定します。

- 作業時間の記録 作業した時間を記録します。
- 進捗率 進捗率を設定します。

バグが発生した場合は、トラッカー「Bug」でチケットを作成します。
課題が発生した場合は、トラッカー「Issue」でチケットを作成します。
開始日、期日、予定工数は、スクラムマスターが設定します。

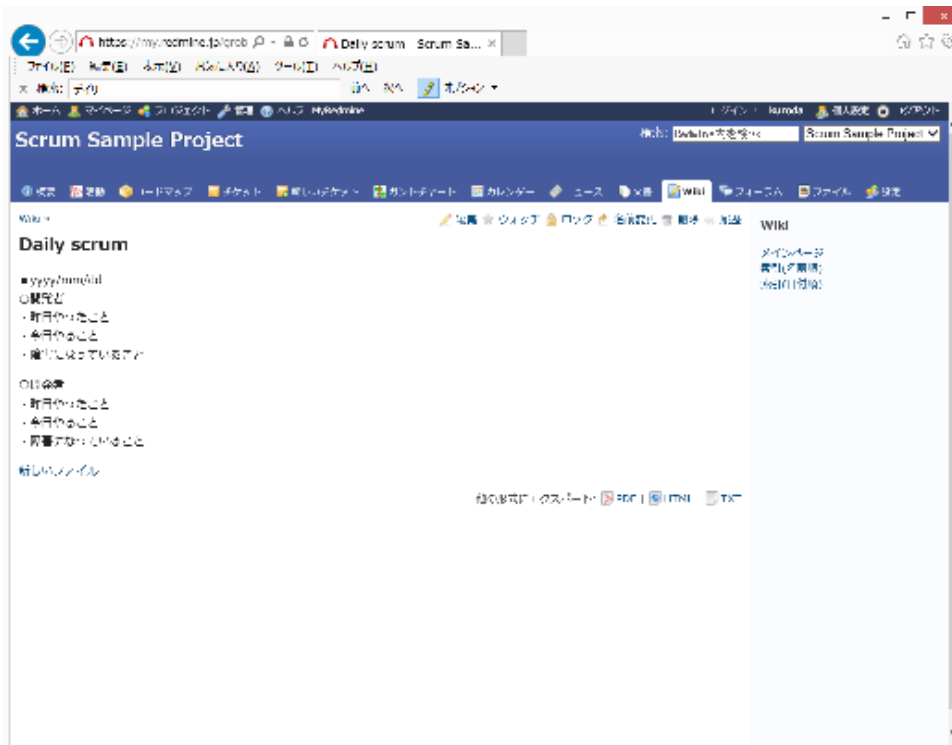
17

9.4 スプリント(デイリースクラム)

デイリースクラムを行い、結果をWikiに公開します。

全員で進捗を確認するミーティングです。

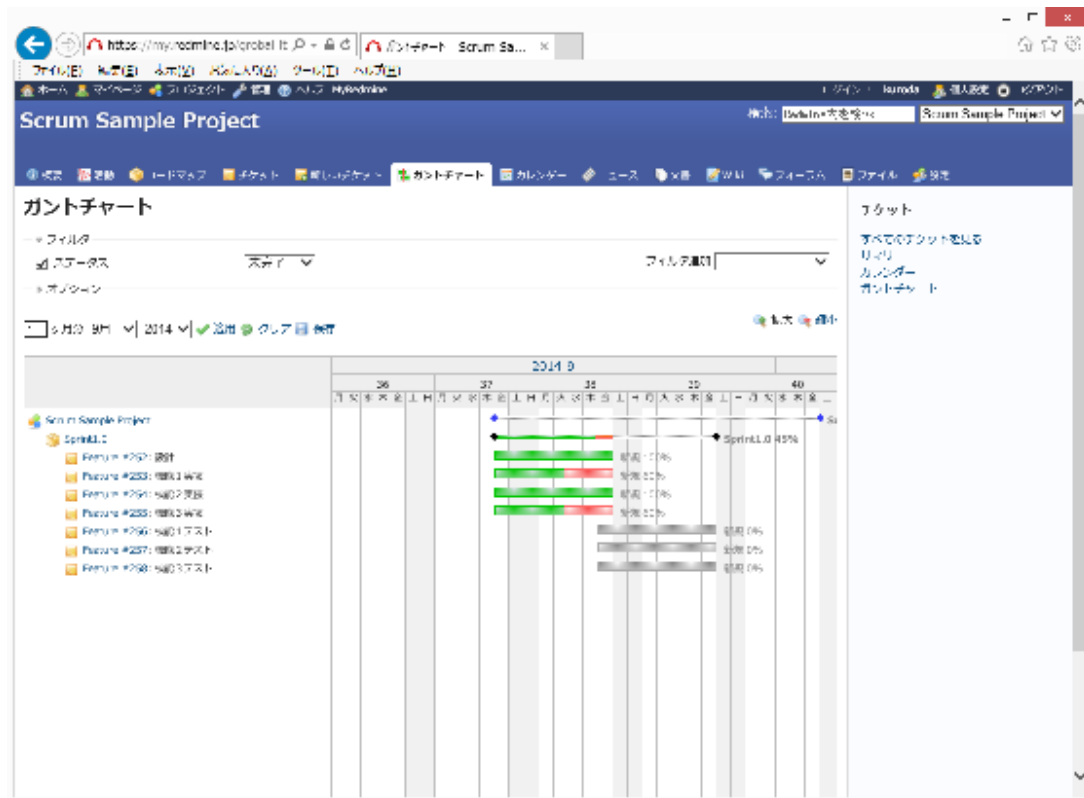
「昨日やったこと」、「今日やること」、「障害になっていること」を共有し



18

9.5 スプリント(進捗管理)

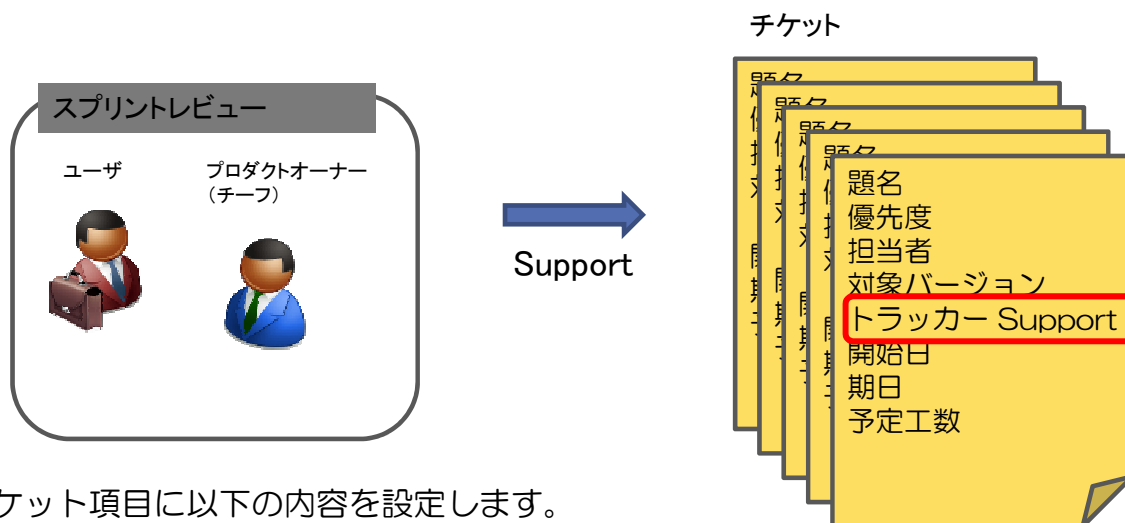
ガントチャートで進捗管理します。
予定工数、作業時間、進捗からガントチャートが作成されます。



19

9.6 スプリントレビュー

スプリントで開発されたプロダクトのレビューを行います。
必要に応じて改善要望、障害のプロダクトバックログを追加します。

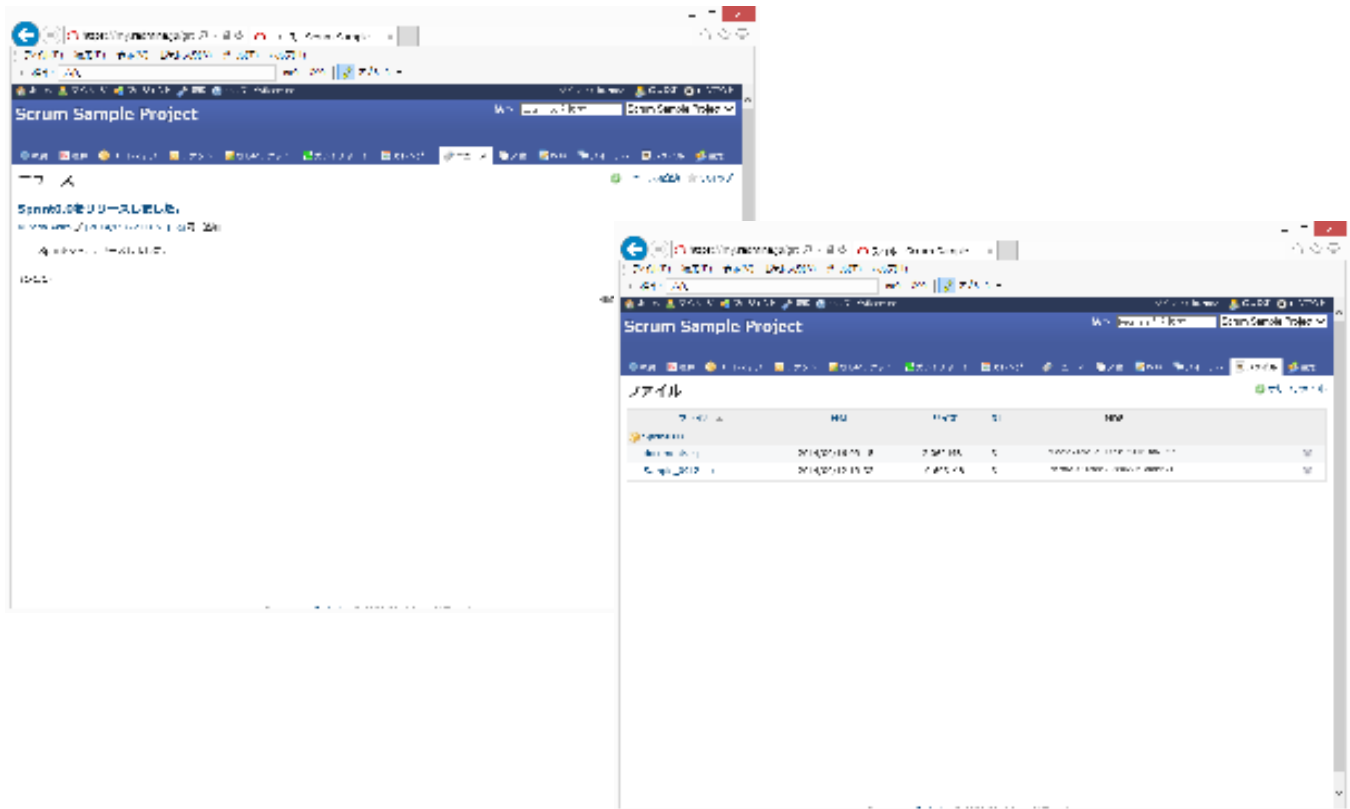


チケット項目に以下の内容を設定します。
・トラッカー Support

20

9.7 スプリントリリース

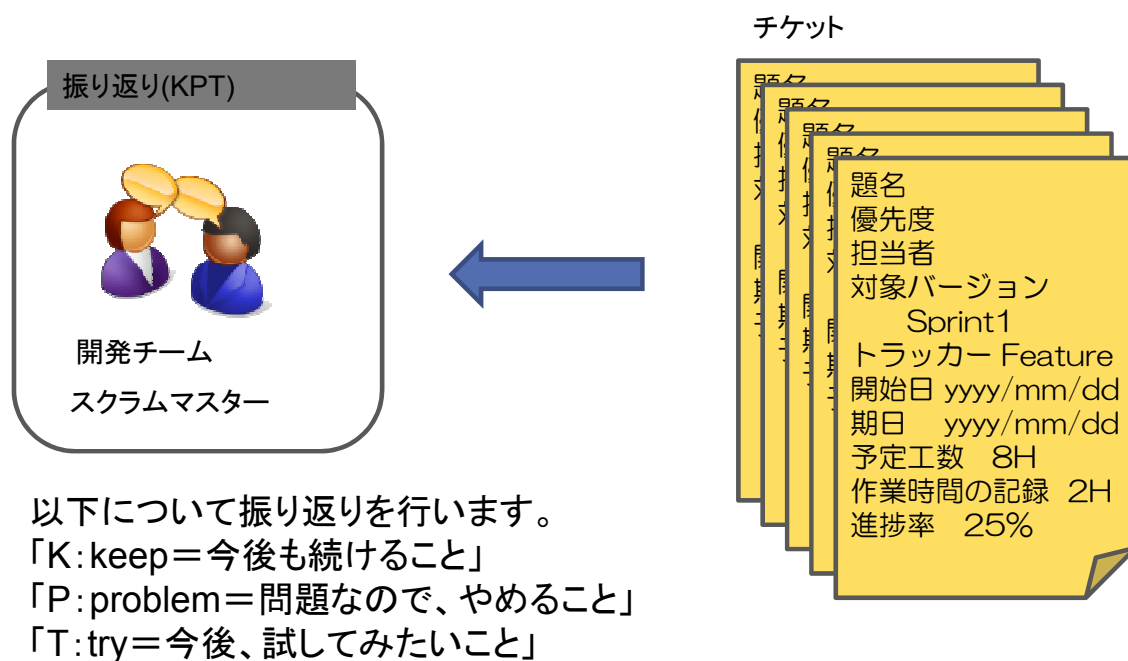
リリース後にニュースやモジュールを公開します。



21

9.8 振り返り(KPT)

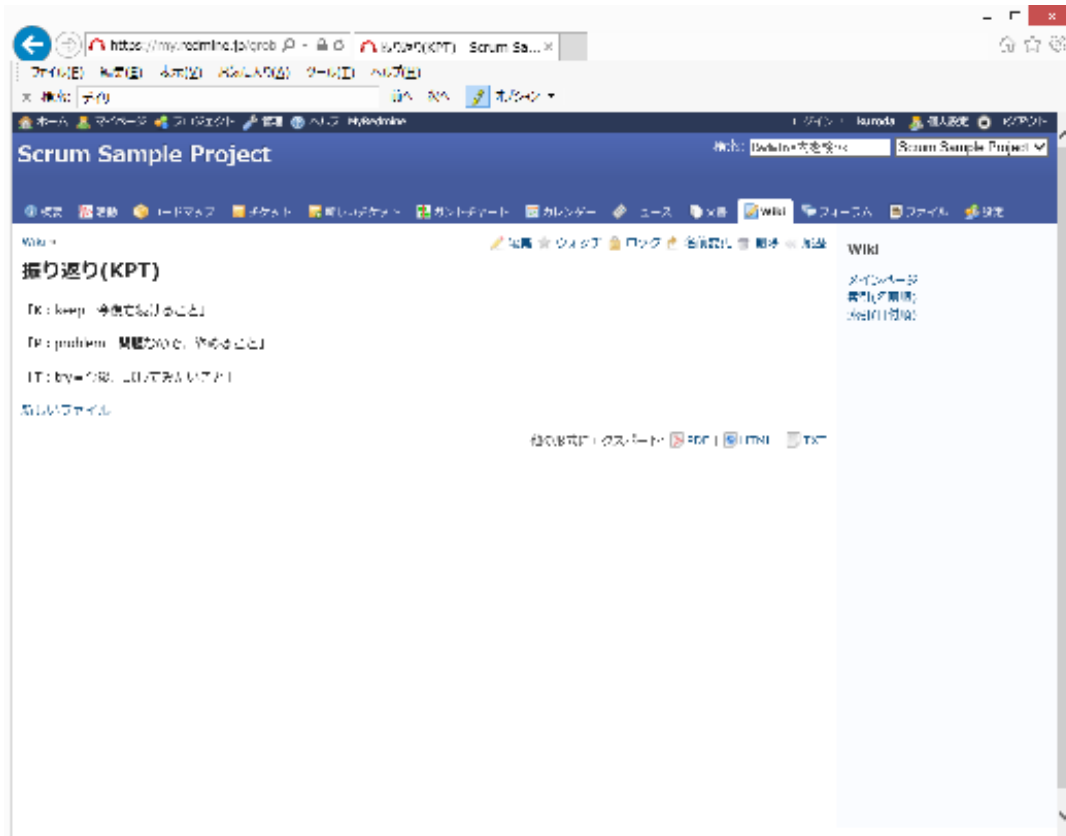
振り返りの判断材料（予定工数、作業時間）をチケットから取得して振り返りを行います。



22

9.9 振り返り(KPT) 情報共有

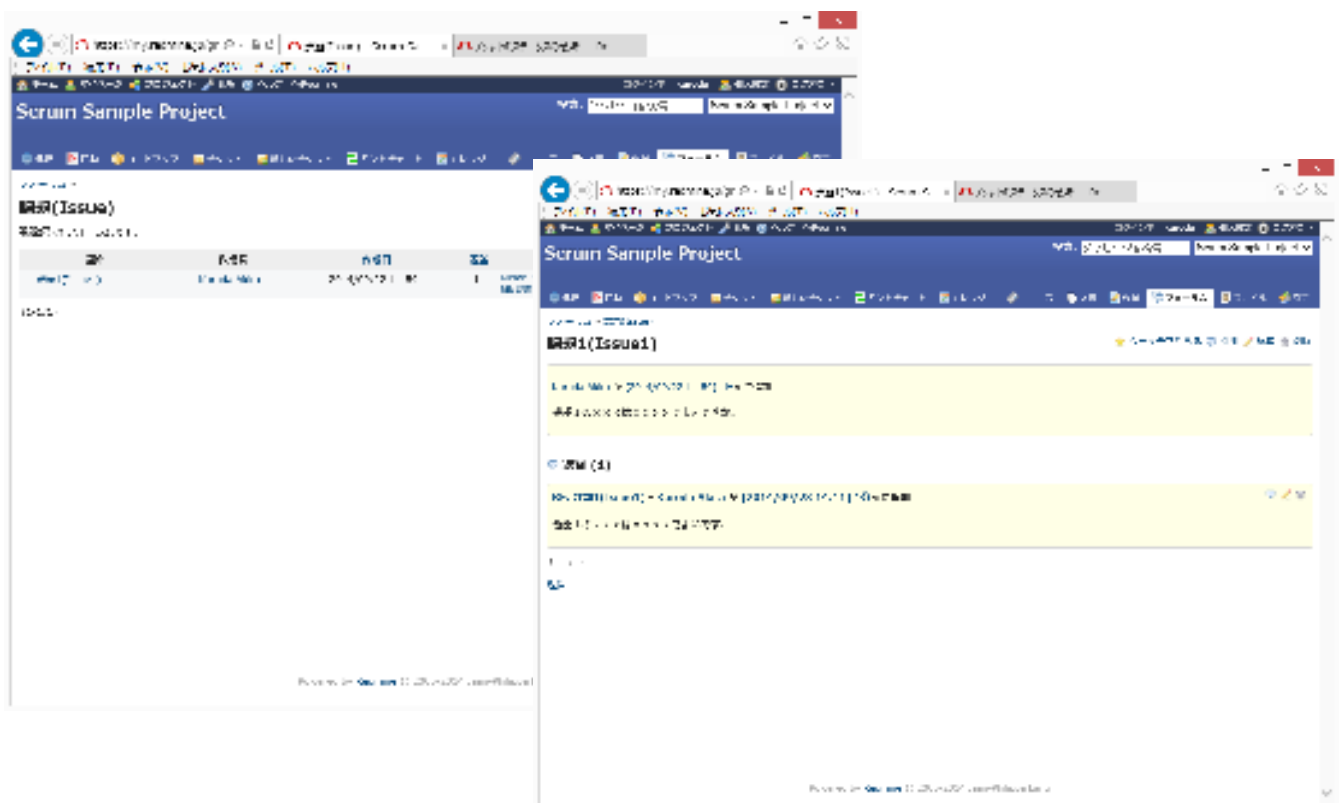
振り返りの結果を、Wikiで情報共有します。



23

9.10 課題の解決方法

Issueチケットを作成した際、機能仕様の不明点、疑問点などの課題をフォーラムで議論し解決します。検討結果は、Issueチケットに記載します。



24

10.1 実証講座プロジェクト情報

プロジェクト情報

サイトURL = <http://~~~~~/>

プロジェクト	参加ユーザ	参加ユーザのロール
チームA	JPN-Usr1	プロジェクトマネージャ
	JPN-Usr2	システムアーキテクト
	JPN-Usr3	プロダクトオーナー（チーフ）
	OffA-Usr1	プロダクトオーナー
	OffA-Usr2	スクラムマスター
	OffA-Usr3	開発者
	OffA-Usr4	開発者
	OffA-Usr5	開発者
チームB	JPN-Usr1	プロジェクトマネージャ
	JPN-Usr2	システムアーキテクト
	JPN-Usr4	プロダクトオーナー（チーフ）
	OffB-Usr1	プロダクトオーナー
	OffB-Usr2	スクラムマスター
	OffB-Usr3	開発者
	OffB-Usr4	開発者
	OffB-Usr5	開発者

※オフショア側チームが2チームで開発を実施する場合の例

25

10.2 実証講座ユーザ情報

ユーザ	パスワード	表示名	メールアドレス	言語
JPN-Usr1	password	JPN Usr1	JPN-Usr1@xxxx.co.jp	日本語
JPN-Usr2	password	JPN Usr2	JPN-Usr2@xxx.co.jp	日本語
JPN-Usr3	password	JPN Usr3	JPN-Usr3@xxx.co.jp	日本語
JPN-Usr4	password	JPN Usr4	JPN-Usr4@xxx.co.jp	日本語
OffA-Usr1	password	OffA Usr1	OffA-Usr1@xxx.co.jp	英語
OffA-Usr2	password	OffA Usr2	OffA-Usr2@xxx.co.jp	英語
OffA-Usr3	password	OffA Usr3	OffA-Usr3@xxx.co.jp	英語
OffA-Usr4	password	OffA Usr4	OffA-Usr4@xxx.co.jp	英語
OffA-Usr5	password	OffA Usr5	OffA-Usr5@xxx.co.jp	英語
OffB-Usr1	password	OffB Usr1	OffB-Usr1@xxx.co.jp	英語
OffB-Usr2	password	OffB Usr2	OffB-Usr2@xxx.co.jp	英語
OffB-Usr3	password	OffB Usr3	OffB-Usr3@xxx.co.jp	英語
OffB-Usr4	password	OffB Usr4	OffB-Usr4@xxx.co.jp	英語
OffB-Usr5	password	OffB Usr5	OffB-Usr4@xxx.co.jp	英語

※初回ログイン時、パスワードを変更してください

26

10.3 ユーザ情報変更方法

表示名、メールアドレス、言語を変更する場合、Redmineにログイン後、以下の画面で変更してください。

The screenshot shows the Redmine web interface. The user is logged in as 'PHD-User1'. The 'My account' page is displayed, with the 'Information' section highlighted by a red box. The fields within this section are:

- First name: [User1]
- Last name: [User1]
- Email: [User1@redmine.jp]
- Language: [English]

Other sections visible on the page include 'Email notifications' (with a dropdown menu set to 'Only for things I watch or I'm involved in'), 'Preferences' (with checkboxes for 'Hide my email address', 'Time zone', 'Display comments', and 'Warn me when leaving a page with unsaved text'), and 'My account' details (showing login name, creation date, and a 'Delete my account' button).

2014 MEXT Project -for Exercise Couse

Implimented function list on Sprint0.

1. Outline

Develop an application, which enables local people and tourists to register and access information about sights in the area.
Users can find the information of registered sights on the maps near the current location.
Users can also search the registered information using suitable search terms.
Moreover, users can vote and evaluate the information offered to the locations.

2. Environment

This application works on latest version Android devices.
For smartphones, this systems requires LTE, 3G or Wi-Fi internet access.
The application transmits data between the private web server of this project.
The applications uses the Google Map to register sights on smartphones.
Agile software and Scrum are used for the software development framework.
Iteration - (called Sprint on Scrum) for one circle is 3 weeks.
We assume that one day = 4.0 hours, 3 weeks =15 days, and 1 Sprint (iteration) for 60 hours.
We plan to work for 2 Sprints (6 weeks in total) on this project.

3. Requirements

3-1 Function Requirements of the application

- Completed (1) When the users start the application, the main menu appears on the screen.
Users can choose the desired functions from the listing on the main menu.
The two main listed functions are: "Sight Information (Map)" and "Rankings".
- Completed (2) When the user start the application for the first time, the registration page opens first to register an email address and password.
The registered information is stored automatically on the application, and it is not required to provide any details on any revisit later.
(The registration page will not be displayed on the second visit)
The registered e-mail and password are required when the user wish to delete the registered information.
- Completed (3) The language of the application depends on the settings of the smartphone.
(The default setting is English)

3-2 Function Requirements on the "sight information (Map)"

- Completed (2) When a sight on the map is chosen, the application will list the detail information of the sight.
The detail information includes the name of the sight, address, phone number, notes and any other required information.
The name of the sight is written both in English and the local language.
Consider to add descriptions on other items in English besides the local language.
- Completed (4) Viewers can leave comments on the detail information page of the sight.
- Completed (5) Users can use keywords and click on the "search" button for full-text search. The search results are listed on the screen.
When the user clicks the data on the list, the display shows the map of the results..
- Completed (6) Clicking "Current Location" button on the display shows the current location on the map.
- Completed (7) Users can register new sights on the map and store the date on the server by long click on the spot.
Users will need their email and password to do this action.

3-3 Function Requirements on the "Rankings"

3-4 Function Requirements on "Registration of email address"

- Completed (1) Users can register and update their email address and password.
A user visits the page for the first time, the registration page appears automatically.
The registered e-mail address and password are stored on the application.

開発機能要件 スプリント0で実装した機能

1. 概要

観光客や地元の人々がその地域の見所を登録して発信するアプリケーションを開発する。
利用者は、現在場所付近の地図上に登録された情報を確認する事が出来る。更にいくつかの条件を入力して、登録情報の検索を行うことが出来る。
また登録されている情報に賛同した利用者が投票する機能を持ち、どの情報が利用者にとって人気があるかも分かる。

2. 環境

利用者は、最新バージョンの 안드로이드 端末(スマートフォン)を使って当アプリケーションを利用する。
スマートフォンは、LTEや3G及びWiFiなどインターネットに接続状態で当アプリケーションを利用する。
当アプリケーションは、予め用意された専用のWebサーバとデータの送受信を行う。
スマートフォンの画面に表示する地図情報は、Google社のMAPを利用する。
開発手法としてアジャイル開発を実施する。また開発フレームワークとしてスクラム(scrum)を使用する。
反復期間(イテレーション:スクラム上ではスプリント(sprint)と呼ぶ)は、3週間で1回とする。
1日=4.0時間、3週間=15日間を想定し、1スプリント(イテレーション)=60時間を想定する。
当プロジェクトにおいて、3スプリント(全9週間)を実施する。

3. 要望事項

3-1 アプリケーション全体に関する要望

- 済 (1) アプリケーションを起動するとメニューが表示され、実行したい機能を選択できる。
現時点では、“情報表示(地図)”と、“各種ランキング”の2機能から選択する。
- 済 (2) 初めて利用する場合、スマートフォンのメールアドレスとパスワードを登録する画面を表示する。
メールアドレスとパスワードは、アプリケーション内に自動的に保管され、2回目以降の利用時は入力する必要は無い。(2回目以降の利用時は、この登録画面は自動表示されない。)
登録されたメールアドレスとパスワードは、本人が登録した情報を削除する場合に利用する。
- 済 (3) 当アプリケーションは、日本語や英語など使用するスマートフォンの言語設定に合わせて画面表示する文字の言語を切替える。(デフォルトは英語表示とする)

3-2 “情報表示(地図)”に関する要望

- 済 (2) 地図上のマークを選択した時に、その地点の詳細情報を画面に表示する。
詳細情報には、名称、住所、電話番号、備考に加えて必要と考えられる項目が表示される。
- 済 (4) 地点の詳細情報表示画面上で、閲覧者がコメントを登録する機能を持つ。
- 済 (5) 検索機能は、全文検索とし画面上でキーワードを入力し、検索ボタンを押すと対象のデータが一覧表示される。
- 済 (6) 現在地ボタンを押すとスマートフォンの現在地を中心とした地図表示を行う。
- 済 (7) 地図表示画面のマークが無い1地点を選択する事(長押し)で、新規に地点を設定し、サーバに登録する機能を持つ。この際、登録者のメールアドレスとパスワードも登録する。

3-3 “各種ランキング”に関する要望

3-4 “メールアドレス登録”に関する要望

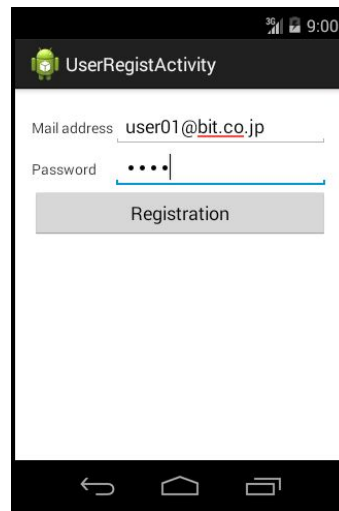
- 済 (1) アプリ利用者のメールアドレスとパスワードを登録・変更できる機能を持つ。
初めて当アプリを起動した時に、自動的に登録画面を表示し、アプリ内にメールアドレスとパスワード情報を保管する。

MapApplication Sample ScreenShots (Base system on the Sprint0)

(1)App icon



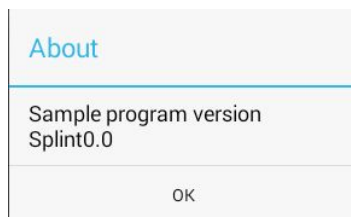
(2)Register user/password



(3)Main Menu



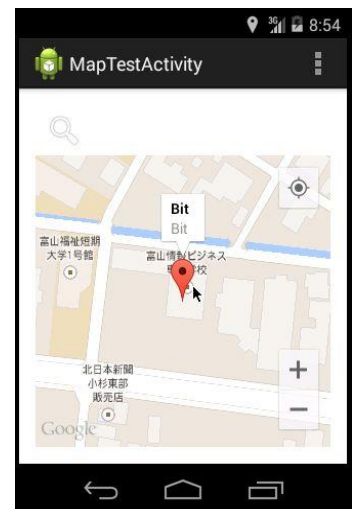
(4)About screen



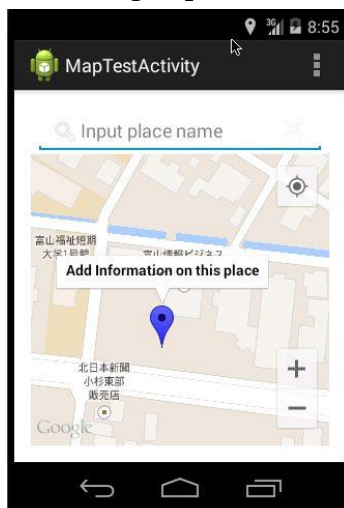
(5)Display Map



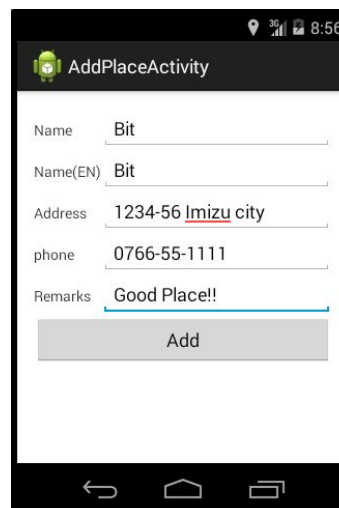
(6)Selected the marker



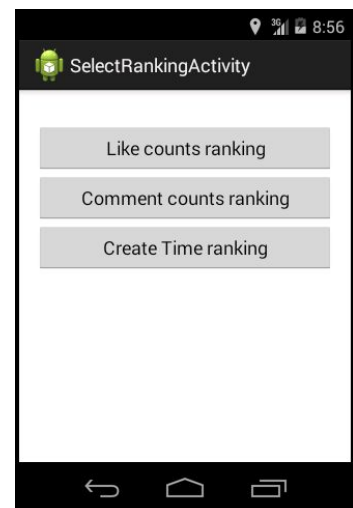
(7)Add new place
(Long tap)



(8)Show detail



(9)Ranking menu

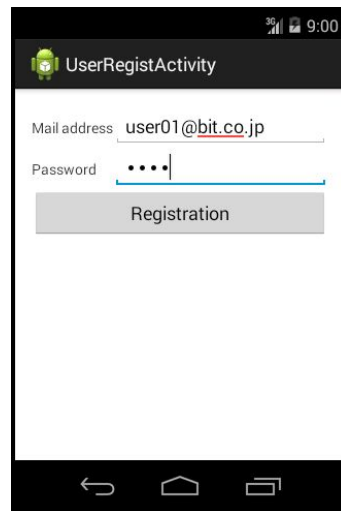


MapApplication サンプル画面 (Sprint0 時点で準備されている画面一覧)

(1)初期アイコン



(2)ユーザ／パスワード登録画面

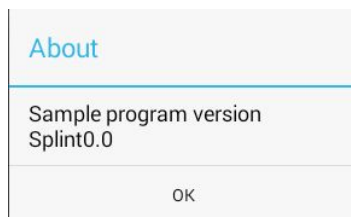


(3)メインメニュー



(4)About 画面表示

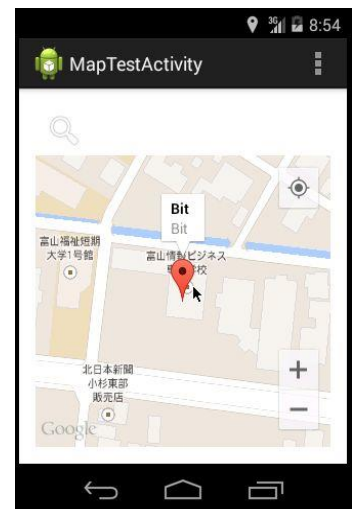
(About ボタンを押した時)



(5)Display Map 表示画面

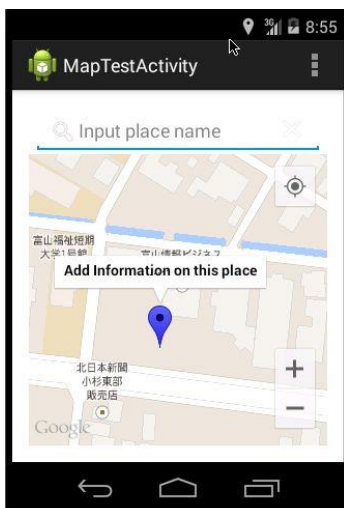


(6)マーカークリック時

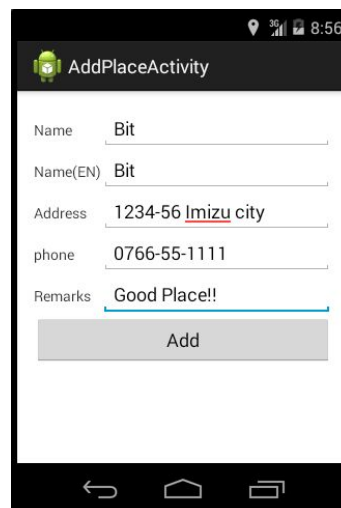


(7)新規登録時

(画面の長押し時)

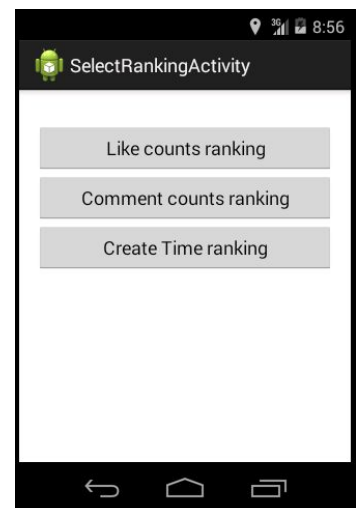


(8)詳細情報新規登録画面



(9)ランキングメニュー

表示画面



1. Outline

Develop an application, which enables local people and tourists to register and access information about sights in the area.
Users can find the information of registered sights on the maps near the current location.
Users can also search the registered information using suitable search terms.
Moreover, users can vote and evaluate the information offered to the locations.

2. Environment

This application works on latest version Android devices.
For smartphones, this systems requires LTE, 3G or Wi-Fi internet access.
The application transmits data between the private web server of this project.
The applications uses the Google Map to register sights on smartphones.
Agile software and Scrum are used for the software development framework.
Iteration - (called Sprint on Scrum) for one circle is 3 weeks.
We assume that one day = 4.0 hours, 3 weeks =15 days, and 1 Sprint (iteration) for 60 hours.
We plan to work for 3 Sprints (9 weeks in total) on this project.

3. Requirements

3-1 Function Requirements of the application

- (1) When the users start the application, the main menu appears on the screen.
Users can choose the desired functions from the listing on the main menu.
The two main listed functions are: "Sight Information (Map)" and "Rankings".
The third function is for the users to update their registered information, such as e-mail address and password.
- (2) When the user start the application for the first time, the registration page opens first to register an email address and password.
The registered information is stored automatically on the application, and it is not required to provide any details on any revisit later.
(The registration page will not be displayed on the second visit)
The registered e-mail and password are required when the user wish to delete the registered information.
- (3) The language of the application depends on the settings of the smartphone.
(The default setting is English)
- (4) The version of the application can be listed from the action bar of the screen.

3-2 Function Requirements on the "sight information (Map)"

- (1) Display the map within one-kilometer radius from the current location and display the mark the registered sights on the server.
Additional options to select the map scale(within 0.5 km, 1.0 km or 2.0 km) will be better.
- (2) When a sight on the map is chosen, the application will list the detail information of the sight.
The detail information includes the name of the sight, address, phone number, notes and any other required information.
The name of the sight is written both in English and the local language.
Consider to add descriptions on other items in English besides the local language.
- (3) Set up "Like" button on the detail information page. Users can vote for the sight by clicking the button.
For a specific sight, only one vote will be accepted from one unique email address, and only the latest vote from the same address is valid.
- (4) Viewers can leave comments on the detail information page of the sight.
Comments are listed by registered dates in descending order
Comment can be deleted by the writer of the comment.
The user who register the sight can also delete the comments.
- (5) Users can use keywords and click on the "search" button for full-text search. The search results are listed on the screen.
When the user clicks the data on the list, the display shows the map of the results..
(Pressing the "back" button on the list page, takes the page back to the map of the current location.)
- (6) Clicking "Current Location" button on the display shows the current location on the map.
- (7) Users can register new sights on the map and store the date on the server by long click on the spot.
Users will need their email and password to do this action.

3-3 Function Requirements on the "Rankings"

(1) Variation of rank order for detail information:

- Rank in order of a number of "Like" votes
- Rank in order of a number of comments
- Rank in order of dates and time

(2) Rank in order of a number of "Like" votes lists the information below:

Items: number of "Like", sight name, address

Sorting order: number of "Like" (descending order), sight names (alphabetical order)

* If possible, the list should be sorted by distance from the current location.

(3) Rank in order of a number of comments lists the information below:

Items: Number of comments, sight name, address

Sorting order: comments (descending order), sight names (alphabetical order)

* If possible, the list will be sorted by distance from the current location.

(4) Rank in order of dates and time

Items: Registered date and time, sight names and address

Sorting order: Registered date and time (descending order), sight names (ascending order)

* If possible, the list will be sorted by distance from the current location.

3-4 Function Requirements on "Registration of email address"

(1) Users can register and update their email address and password.

A user visits the page for the first time, the registration page appears automatically.

The registered e-mail address and password are stored on the application.

(2) We expect the appropriate implementation for the new function to update password.

1. 概要

観光客や地元の人々がその地域の見所を登録して発信するアプリケーションを開発する。
利用者は、現在場所付近の地図上に登録された情報を確認する事が出来る。更にいくつかの条件を入力して、登録情報の検索を行うことが出来る。
また登録されている情報に賛同した利用者が投票する機能を持ち、どの情報が利用者にとって人気があるかも分かる。

2. 環境

利用者は、最新バージョンのアンドロイド端末(スマートフォン)を使って当アプリケーションを利用する。
スマートフォンは、LTEや3G及びWiFiなどインターネットに接続状態で当アプリケーションを利用する。
当アプリケーションは、予め用意された専用のWebサーバとデータの送受信を行う。
スマートフォンの画面に表示する地図情報は、Google社のMAPを利用する。
開発手法としてアジャイル開発を実施する。また開発フレームワークとしてスクラム(scrum)を使用する。
反復期間(イテレーション: スクラム上ではスプリント(sprint)と呼ぶ)は、3週間で1回とする。
1日=4.0時間、3週間=15日間を想定し、1スプリント(イテレーション)=60時間を想定する。
当プロジェクトにおいて、3スプリント(全9週間)を実施する。

3. 要望事項

3-1 アプリケーション全体に関する要望

- (1) アプリケーションを起動するとメニューが表示され、実行したい機能を選択できる。
現時点では、“情報表示(地図)”と、“各種ランキング”の2機能から選択する。
また3番目の機能として端末登録情報(メールアドレスとパスワード)の変更機能を有する。
- (2) 初めて利用する場合、スマートフォンのメールアドレスとパスワードを登録する画面を表示する。
メールアドレスとパスワードは、アプリケーション内に自動的に保管され、2回目以降の利用時は入力する必要は無い。(2回目以降の利用時は、この登録画面は自動表示されない。)
登録されたメールアドレスとパスワードは、本人が登録した情報を削除する場合に利用する。
- (3) 当アプリケーションは、日本語や英語など使用するスマートフォンの言語設定に合わせて画面表示する文字の言語を切替える。(デフォルトは英語表示とする)
- (4) 画面のアクションバーから、アプリケーションのバージョン情報表示を行える。

3-2 “情報表示(地図)”に関する要望

- (1) 当機能を表示した際にスマートフォンの現在地情報より、現在地付近(半径1km以内)の地図を表示し、サーバ上に情報が登録されている地点にマークを表示する。
付近地の範囲(0.5Km、1.0Km、2.0Kmなど)を選択できるようになると更に良い。
- (2) 地図上のマークを選択した時に、その地点の詳細情報を画面に表示する。
詳細情報には、名称、住所、電話番号、備考に加えて必要と考えられる項目が表示される。
名称項目は、英語表記と現地語表記の2種類を持ち、その他の項目については、必要に応じて現地語表記に加えて、英語表記を追加する事を検討。
- (3) 地点の詳細情報表示画面には、“いいね”(Like)ボタンが置かれて、このボタンをクリックする事でその地点に1票を登録する事ができる機能を持つ。
但し、1メールアドレスで1回のみ投票(最新の投票のみ有効)になるように機能する。
- (4) 地点の詳細情報表示画面上で、閲覧者がコメントを登録する機能を持つ。
コメントは登録日時の降順で表示する。
コメントはコメントの登録者本人であれば、削除可能とする。
コメントはその地点情報の登録者も削除可能とする。
- (5) 検索機能は、全文検索とし画面上でキーワードを入力し、検索ボタンを押すと対象のデータが一覧表示される。一覧表示からデータをクリックするとその登録地点を中心とした地図を表示する。(一覧表示で戻るボタンを押すと地図表示に戻る)
- (6) 現在地ボタンを押すとスマートフォンの現在地を中心とした地図表示を行う。
- (7) 地図表示画面のマークが無い1地点を選択する事(長押し)で、新規に地点を設定し、サーバに登録する機能を持つ。この際、登録者のメールアドレスとパスワードも登録する。

3-3 “各種ランキング”に関する要望

- (1) ランキングとして以下の3種類の機能を持つ。
 - ・ “いいね”の多い情報ランキング
 - ・ コメント数の多い情報ランキング

・登録日時順ランキング

- (2) ”いいね”の多い情報ランキングは、以下の情報を一覧表示する。

項目: ”いいね”獲得数、名称、住所

並び順: ”いいね”獲得数(降順)、名称(昇順)

※ 可能であれば、現在位置から近い順番も考慮する。

- (3) コメント数の多い情報ランキングは、以下の情報を一覧表示する。

項目: コメント数、名称、住所

並び順: コメント数(降順)、名称(昇順)

※ 可能であれば、現在位置から近い順番も考慮する。

- (4) 登録日時順ランキング

項目: 登録日時、名称、住所

並び順: 登録日時(降順)、名称(昇順)

※ 可能であれば、現在地から近い順番も考慮する。

3-4 ”メールアドレス登録”に関する要望

- (1) アプリ利用者のメールアドレスとパスワードを登録・変更できる機能を持つ。

初めて当アプリを起動した時に、自動的に登録画面を表示し、アプリ内にメールアドレスとパスワード情報を保管する。

- (2) 情報を登録した後で、パスワードを変更する場合の動作については、より良い実装方法の提案を期待する。

Database Specification

1 Relationship diagram

The relationship diagram in database is shown in fig.1

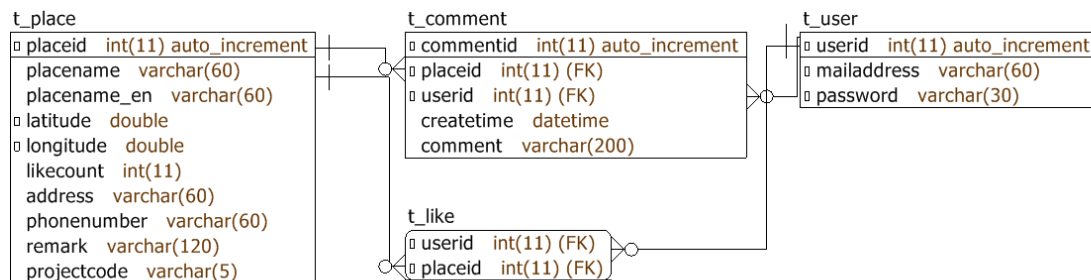


図 1 Relationship diagram

2 Database table Specification

2.1 Place information table

This table stores information related to the place, e.g. the place name, latitude, longitude, and etc.

Table name	t_place				
Item	Column name	Data type	PK	NOT NULL	Remarks
Place ID	placeid	Int	○	○	Auto-increment
Place name	placename	Varchar(60)			
Place name (English)	placename_en	Varchar(60)			
Latitude	latitude	Double		○	
Longitude	longitude	Double		○	
Like counts	likecount	Int			Default sets '0'
Address	address	Varchar (60)			
Phone number	phonenumber	Varchar (15)			Do not have to include Hyphen
Remark	remark	Varchar (120)			
Project code	projectcode	Varchar (5)			Sample program uses "JPN00"

2.2 User table

This table stores user registration information.

Table name	t_user				
Item	Column name	Data type	PK	NOT NULL	Remarks
User ID	userid	Int	○	○	Auto-increment
Mail address	mailaddress	Varchar (60)		○	
Password	password	Varchar (30)		○	

2.3 Comment table

This table stores the comment registered each place.

Table name	t_comment				
Item	Column name	Data type	PK	NOT NULL	Remarks
Comment ID	commentid	Int	○	○	Auto-increment
Place ID	placeid	Int		○	
User ID	userid	Int		○	
Comment date and time	commenttime	datetime			
Comment text	comment	Varchar(200)			

2.4 Like count manage table

This table manages the Place ID and the user who made the vote by 'Like' button.

Table name	t_like				
Item	Column name	Data type	PK	NOT NULL	Remarks
Place ID	placeid	Int	○	○	
User ID	userid	Int	○	○	

データベーステーブル仕様書

1 テーブル関連図

本システムのデータベーステーブル関連図を図 1 に示す。

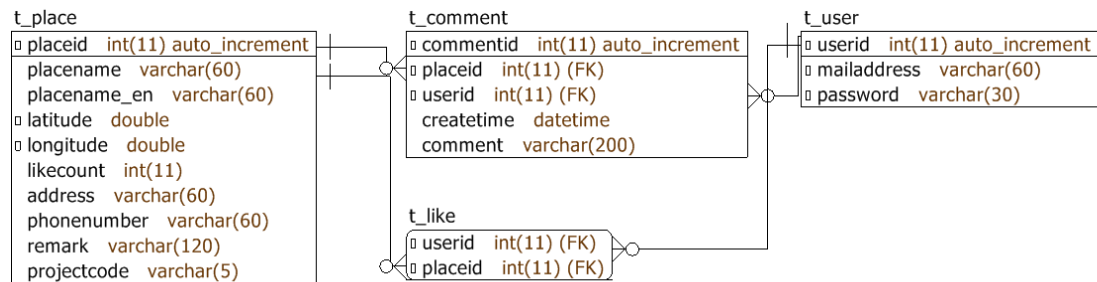


図 1 データベーステーブル関連図

2 テーブル仕様

2.1 地点情報テーブル

地点の緯度経度、および詳細情報を記録する。

テーブル名	t_place				
項目	カラム名	型	PK	必須	備考
地点 ID	placeid	Int	○	○	登録時に自動で加算される
地点名称	placename	Varchar(60)			
地点名称 (英語)	placename_en	Varchar(60)			
緯度	latitude	Double		○	
経度	longitude	Double		○	
いいね数	likecount	Int			初期値は"0"
住所	address	Varchar (60)			
電話番号	phonenumber	Varchar (15)			ハイフン不要
備考	remark	Varchar (120)			
プロジェクト コード	projectcode	Varchar (5)			サンプルデータは「JPN00」

2.2 ユーザテーブル

端末の登録情報を保持する。

テーブル名	t_user				
項目	カラム名	型	PK	必須	備考
ユーザ ID	userid	Int	○	○	登録時に自動で加算される
メールアドレス	mailaddress	Varchar (60)		○	
パスワード	password	Varchar (30)		○	

2.3 コメントテーブル

地点毎に登録されたコメントを記録する。

テーブル名	t_comment				
項目	カラム名	型	PK	必須	備考
コメント ID	commentid	Int	○	○	登録時に自動で加算される
地点 ID	placeid	Int		○	
ユーザ ID	userid	Int		○	
コメント日時	commenttime	datetime			
コメント	comment	Varchar(200)			

2.4 いいね(Like)管理テーブル

“いいね”(Like)ボタンで投票を行われた地点とユーザを記録する。

テーブル名	t_like				
項目	カラム名	型	PK	必須	備考
地点 ID	placeid	Int	○	○	
ユーザ ID	userid	Int	○	○	

Interface specifications between server and Android

1 Messages

Messages between server and mobile phone are shown in Table.1.

Table 1. Messages between server and Android

Message	Transfer Protocol	Function
Search place information	HTTP	Search the registered place information by place name
Register place information	HTTP	Register place information from client
Delete place information	HTTP	Delete place information from client
Register user information	HTTP	Register user information
Delete user information	HTTP	Delete place information
Search comment	HTTP	Search comments of the particular place information
Register comment	HTTP	Register a comment to the place information
Delete comment	HTTP	Delete a comment of the particular place information

2 Message specifications

The function of request from an android device use HTTP post method.

The response from the server is made of HTTP response header and body with JSON format.

2.1 Search place information

Set keyword and throw the request to server. Then get the place information data which are only matched data on the database field. If you set blank on keyword then you get all place information data.

○Request of “Search place information”

Protocol	HTTP POST request	
URL	http://mail.tym-bit.com/SearchPlace.php	
Parameters	Attr.ID	Note
Keyword text	keyword	The keyword to search from place name field on the table. If Keyword is blank then you get all data on the table.
Project code	pj_code	Set fixed project code for your project.

○Response of “Search place information” (Normal)

Format	HTTP response header + body with JSON object	
Status code	200 (OK)	
JSON object	Attr.ID	Note
Number of data	count	Number of result
Array of place information	data	JSON object array of the place information as follows:
Place information	Attr. ID	Note
Place ID	placeid	Unique key of this data
Place Name	placename	Place name in local language
Place Name (English)	placename_en	Place name in English
Latitude	latitude	Latitude of this place
Longitude	longitude	Longitude of this place
Number of Like	likecount	Number of Like that users clicked
Address	address	
Phone	phonenum ber	
Remarks	remark	
Project code	projectcode	The fixed unique value of each project

○Response of “Search place information” (Error)

Format	HTTP response header + body with JSON object	
Status code	500 (Internal Server Error)	
JSON Object	Attr. ID	Note
Error code	errcode	Error code is occurred in MySQL
Error message	message	Error message text from MySQL

2.2 Register place information

Add a new place information to database

○Request of “Register place information”

Protocol	HTTP POST request	
URL	http://mail.tym-bit.com/AddPlace.php	
Parameters	Attr. ID	Note
Place name	placename	Place name in local language
Place name	placename_en	Place name in English
Latitude	latitude	(mandatory)
Longitude	longitude	(mandatory)
Address	address	(optionally)
Phone	phonenumber	(optionally)
Remarks	remark	(optionally)
Project code	projectcode	The fixed unique value of each project

○Response of “Register place information” (Normal)

Format	HTTP response header + body with JSON object	
Status code	200 (OK)	
JSON object	Attr. ID	Note
Place ID	placeid	Place ID is created automatically

○Response of “Register place information” (Error)

Format	HTTP response header + body with JSON object	
Status code	500 (Internal Server Error)	
JSON object	Attr. ID	Note
Error code	errcode	Error code is occurred in MySQL
Error message	message	Error message text from MySQL

2.3 Delete place information

Delete a place information with Place ID parameter.

○Request of “Delete place information”

Protocol	HTTP POST request	
URL	http://mail.tym-bit.com/DeletePlace.php	
Parameters	Attr. ID	Note
Place ID	placeid	Place ID of the target place information

○Response of “Delete place information” (Normal)

Format	HTTP response header
Status code	200 (OK)

○Response of “Delete place information” (Error)

Format	HTTP response header+JSON object	
Status code	500 (Internal Server Error)	
JSON object	Attr. ID	Note
Error code	errcode	Error code is occurred in MySQL
Error message	message	Error message text from MySQL

2.4 Register user information

Web Server checks duplicated mail address. If there are same mail addresses, web server returns error.

○Request of “Register user information”

Protocol	HTTP POST request	
URL	http://mail.tym-bit.com/UserRegistration.php	
Parameters	Attr. ID	Note
Mail address	mailaddress	
Password	password	

○Response of “Register user information” (Normal)

Format	HTTP response header+JSON object	
Status code	200 (OK)	
JSON object	Attr. ID	Note
User ID	userid	User ID is created automatically

○Response of “Register user information” (Error)

Format	HTTP response header+JSON object	
Status code	500 (Internal Server Error)	
JSON object	Attr. ID	Note
Error code	errcode	Error code is occurred in MySQL
Error message	message	Error message text from MySQL

2.5 Delete user information

Web server deletes user data from database.

○Request of “Delete user information”

Protocol	HTTP POST request	
URL	http://mail.tym-bit.com/DeleteUser.php	
Parameters	Attr. ID	Note
User ID	userid	

○Response of “Delete user information” (Normal)

Format	HTTP response header	
Status code	200 (OK)	

○Response of “Delete user information” (Error)

Format	HTTP response header+JSON object	
Status code	500 (Internal Server Error)	
JSON object	Attr. ID	Note
Error code	errcode	Error code is occurred in MySQL
Error message	message	Error message text from MySQL

2.6 Search comment

- Get comments of the particular place information. Comment data sequence is sorted by entry date in descending order.

○Request of “Search comment”

Protocol	HTTP POST request	
URL	http://mail.tym-bit.com/SearchComment.php	
Parameters	Attr. ID	Note
Place ID	placeid	Place ID for getting comments

○Response of “Search comment” (Normal)

Format	HTTP response header+JSON object	
Status code	200 (OK)	
JSON object	Attr. ID	Note
Number of data	count	Number of result
Array of comment data	data	JSON object array of the comment data as follows:
Comment data	Attr. ID	Note
Place ID	placeid	
User ID	userid	User ID of commented
Comment Date and Time	createtime	
Comment Text	comment	

○Response of “Search comment” (Error)

Format	HTTP response header+JSON object	
Status code	500 (Internal Server Error)	
JSON object	Attr. ID	JSON object
Error code	errcode	Error code
Error message	message	Error message

2.7 Register comment

Add a new comment to particular place information.

○Request of “Register comment”

Protocol	HTTP POST request	
URL	http://mail.tym-bit.com/AddComment.php	
Parameters	Attr. ID	Note
Place ID	placeid	The place ID adding comment
User ID	userid	
Comment date and time	createtime	
Comment text	comment	

○Response of “Register comment” (Normal)

Format	HTTP response header+JSON object	
Status code	200 (OK)	
JSON object	Attr. ID	Note
Comment ID	commentid	Comment ID is created automatically

○Response of “Register comment” (Error)

Format	HTTP response header+JSON object	
Status code	500 (Internal Server Error)	
JSON object	Attr. ID	JSON object
Error code	errcode	Error code
Error message	message	Error message

2.8 Delete comment

○Request of “Delete comment”

Protocol	HTTP POST request	
URL	http://mail.tym-bit.com/DeleteComment.php	
Parameters	Attr. ID	Note
Comment ID	commentid	

○Response of “Delete comment” (Normal)

Format	HTTP response header	
Status code	200 (OK)	

○コメント登録応答（異常時）

Format	HTTP response header+JSON object	
Status code	500 (Internal Server Error)	
JSON object	Attr. ID	JSON object
Error code	errcode	Error code
Error message	message	Error message

サーバ端末間インタフェース仕様書

1 メッセージ関連図

サーバと携帯端末間でのメッセージ一覧を表 1 に示す。

表 1 メッセージ一覧

メッセージ	プロトコル	内容
地点情報検索	HTTP	登録されている地点情報を名称で検索し取得する
地点情報登録	HTTP	地点情報を登録する
地点情報削除	HTTP	地点情報を登録する
ユーザ情報登録	HTTP	ユーザ情報を登録する
ユーザ情報削除	HTTP	ユーザ情報を削除する
コメント検索	HTTP	地点情報に付加されているコメントを取得する
コメント登録	HTTP	登録されている地点情報にコメントを登録する
コメント削除	HTTP	地点情報に付加されているコメントを削除する

2 メッセージ仕様

端末からサーバへの要求は HTTP POST リクエストで行う。

要求に対するサーバから端末への応答は、HTTP レスポンスヘッダと JSON フォーマットの本文データの組み合わせで返される。

2.1 地点情報検索

検索キーワードに部分的に、もしくは完全に一致する地点名称の地点情報一覧を取得する。検索キーワードが空文字の場合、地点情報の全データを取得する。

○地点情報検索 結果要求

プロトコル	HTTP POST リクエスト	
URL	http://mail.tym-bit.com/SearchPlace.php	
項目	属性 ID	備考
検索キーワード	keyword	検索する地点名称条件を設定できる 空文字のときは条件なし（全件取得）
プロジェクトコード	pj_code	プロジェクト毎に設定するコード、設定必須

○地点情報検索 結果応答（正常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
HTTP ヘッダ ステータスコード	200 (OK)	
JSON オブジェクト 項目	属性 ID	備考
取得件数	count	検索結果データ件数
地点情報データ の検索結果	data	下記の「地点情報データ」を要素に持つ JSON オブジェクト配列
「地点情報データ」 項目名	属性 ID	備考
地点 ID	placeid	地点ごとに固有の番号
地点名称	placename	検索で該当した地点の名称（現地語表記）
地点名称（英語）	placename_ en	検索で該当した地点の名称（英語表記）
緯度	latitude	検索で該当した地点の緯度
経度	longitude	検索で該当した地点の経度
いいね件数	likecount	いいね(Like)ボタンが押された回数
住所	address	
電話番号	phonenumb er	
備考	remark	
プロジェクトコード	projectcode	プロジェクト毎に指定するコード

○地点情報検索 結果応答（異常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	500 (Internal Server Error)	
JSON オブジェクト 項目	属性 ID	備考
エラーコード	errcode	SQLSTATE エラーコード
エラーメッセージ	message	サーバで出力されたエラーメッセージ全文

2.2 地点情報登録

地点情報をデータベースに登録する。

○地点情報登録要求

プロトコル	HTTP POST リクエスト	
URL	http://mail.tym-bit.com/AddPlace.php	
項目	属性 ID	備考
地点名称	placename	地点名称（現地語表記）
地点名称（英）	placename_en	地点名称（英語表記）
緯度	latitude	追加する地点の緯度、設定必須
経度	longitude	追加する地点の経度、設定必須
住所	address	追加する地点の住所（任意設定）
電話番号	phonenumber	追加する地点の電話番号（任意設定）
備考	remark	備考（任意設定）
プロジェクトコード	projectcode	プロジェクトごとに指定するコード 設定必須

○地点情報登録結果応答（正常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	200 (OK)	
JSON オブジェクト 項目	属性 ID	備考
登録した地点 ID	placeid	登録した地点の地点 ID

○地点情報登録結果応答（異常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	500 (Internal Server Error)	
JSON オブジェクト 項目	属性 ID	備考
エラーコード	errcode	SQLSTATE エラーコード
エラーメッセージ	message	サーバで発生したエラーメッセージ

2.3 地点情報削除

地点 ID を送信することで、地点情報を削除する。正常に削除された場合はステータスコード 200(OK)のみを返し、JSON オブジェクトは返さない。

○地点情報削除要求

プロトコル	HTTP POST リクエスト	
URL	http://mail.tym-bit.com/DeletePlace.php	
項目	属性 ID	備考
地点 ID	placeid	登録されている地点 ID

○地点情報削除応答（正常時）

データ形式	HTTP レスポンスヘッダ
ステータスコード	200 (OK)

○地点情報削除応答（異常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	500 (Internal Server Error)	
JSON オブジェクト 項目	属性 ID	備考
エラーコード	errcode	SQLSTATE エラーコード
エラーメッセージ	message	サーバで発生したエラーメッセージ

2.4 ユーザ情報登録

メールアドレスとパスワードを送信することで、ユーザ情報を登録する。サーバはメールアドレスに重複がないか確認し、なければユーザ ID を発行し登録を行い、ユーザ ID を応答として返す。

○ユーザ情報登録要求

プロトコル	HTTP POST リクエスト	
URL	http://mail.tym-bit.com/UserRegistration.php	
項目	属性 ID	備考
メールアドレス	mailaddress	登録するメールアドレス
パスワード	password	登録するパスワード

○ユーザ情報登録応答（正常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	200 (OK)	
JSON オブジェクト 項目	属性 ID	備考
ユーザ ID	userid	登録されたユーザ ID

○ユーザ情報登録応答（異常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	500 (Internal Server Error)	
JSON オブジェクト 項目	属性 ID	備考
エラーコード	errcode	SQLSTATE エラーコード
エラーメッセージ	message	サーバで発生したエラーメッセージ

2.5 ユーザ情報削除

ユーザ ID を送信することで、ユーザ情報を削除する。正常に削除された場合はステータスコード 200(OK)のみを返し、JSON オブジェクトは返さない。

○ユーザ情報削除要求

プロトコル	HTTP POST リクエスト	
URL	http://mail.tym-bit.com/DeleteUser.php	
項目	属性 ID	備考
ユーザ ID	userid	登録されているユーザ ID

○ユーザ情報削除応答（正常時）

データ形式	HTTP レスポンスヘッダ
ステータスコード	200 (OK)

○ユーザ情報削除応答（異常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	500 (Internal Server Error)	
JSON オブジェクト 項目	属性 ID	備考
エラーコード	errcode	SQLSTATE エラーコード
エラーメッセージ	message	サーバで発生したエラーメッセージ

2.6 コメント検索

特定の地点情報のコメントを取得する。地点 ID を条件として渡す。

検索結果データはコメントされた日の降順で渡される。

○コメント検索要求

プロトコル	HTTP POST リクエスト	
URL	http://mail.tym-bit.com/SearchComment.php	
項目	属性 ID	備考
地点 ID	placeid	コメントを検索する地点 ID

○コメント検索 結果応答（正常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
HTTP ヘッダ ステータスコード	200 (OK)	
JSON オブジェクト 項目	属性 ID	備考
取得件数	count	検索結果データ件数
コメントデータ の検索結果	data	下記の「コメントデータ」を要素に持つ JSON オブジェクト配列
「コメントデータ」 項目名	属性 ID	備考
地点 ID	placeid	地点 ID
地点名称	userid	コメントしたユーザ ID
コメント日時	createtime	コメントされた日時
コメント	comment	コメント本文

○コメント検索応答（異常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	500 (Internal Server Error)	
JSON オブジェクト 項目	属性 ID	備考
エラーコード	errcode	SQLSTATE エラーコード
エラーメッセージ	message	サーバで発生したエラーメッセージ

2.7 コメント登録

特定の地点情報に対するコメントをデータベースに登録する。サーバは登録時に発行したコメント ID を応答として返す。

○コメント登録要求

プロトコル	HTTP POST リクエスト	
URL	http://mail.tym-bit.com/AddComment.php	
項目	属性 ID	備考
地点 ID	placeid	コメントする地点 ID
ユーザ ID	userid	コメントするユーザ ID
作成時間	createtime	コメントする時間
コメント	comment	コメント本文

○コメント登録応答（正常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	200 (OK)	
JSON オブジェクト項目	属性 ID	備考
登録したコメント ID	commentid	登録したコメント ID

○コメント登録応答（異常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	500 (Internal Server Error)	
JSON オブジェクト項目	属性 ID	備考
エラーコード	errcode	SQLSTATE エラーコード
エラーメッセージ	message	サーバで発生したエラーメッセージ

2.8 コメント削除

特定のコメントを削除する。コメントしたユーザとその時間の情報を持つ。正常に登録された場合はステータスコード 200(OK)のみを返し、JSON オブジェクトは返さない。

○コメント登録要求

プロトコル	HTTP POST リクエスト	
URL	http://mail.tym-bit.com/DeleteComment.php	
項目	属性 ID	備考
コメント ID	commentid	削除するコメント ID

○コメント登録応答（正常時）

データ形式	HTTP レスポンスヘッダ
ステータスコード	200 (OK)

○コメント登録応答（異常時）

データ形式	HTTP レスポンスヘッダ+JSON オブジェクト	
ステータスコード	500 (Internal Server Error)	
JSON オブジェクト 項目	属性 ID	備考
エラーコード	errcode	SQLSTATE エラーコード
エラーメッセージ	message	サーバで発生したエラーメッセージ