

Created : 2016/03/25

Author : Tanaka Yuki

Updated : 2016/03/31

## Purpose of this catalog

現在の背景に、グローバル開発プロセスでは、案件の管理とフロー管理が実現できていないという現状がある

ハード面では、プラットフォームの整備、ソフト面では、案件管理フローを制定、を行うことでわれわれの業務の入り口である案件の起票、開発・評価フロー、出荷までの一連の案件管理を行いたい

また、この取組を土台にし、より細かな開発フローを今後取り決めていきたい

## Current problem

発見された不具合や新規開発の案件管理を行う場所がないため、下記のような問題をはらんでいる

- ・コンサルタントが開発者に直接連絡を取っており、やり取りが残っていない
- ・対応の優先度等が属人的に決められている
- ・全案件の把握が困難

また、開発フローにおいても

- ・進捗の管理が曖昧
- ・出荷可能になるまでのToDoなどが管理されていない
- ・提出物管理がされていない

の問題を抱えている

## Summary

1. 場所
2. 案件登録/不具合報告のフロー
3. 開発フロー管理方法
4. 今後の展望
5. 課題

## 1. 案件を管理する場所

## 1-1 開発案件／不具合を登録する場所

## 《求められること》

お客様が抱えているトラブルをできる限り詳しい情報とともに登録できる

## 《誰が使う》

Consultant

## 《場所》

「Global@SUPPORT」 <https://support.worksap.com/>

## 《メリット》

- ・お客様とのコミュニケーションの中から案件を起票することができる
- ・案件や指摘不具合の検索が容易（お客様別、優先度別、日付別、...）

## 1-2 開発案件／不具合のチケット管理・進捗管理を行う場所

## 《求められること》

各案件の進捗管理ができ、出荷に必要な情報を管理できる

## 《誰が使う》

Developer & Evaluator

## 《場所》

「(仮)Trac for CPNYGP」 （現在未設置）

Tracは、ATEが開発・保守をしているPRM([http://sho-gouki/trac/trac\\_dev/wiki](http://sho-gouki/trac/trac_dev/wiki))をベースに

CJKが現在運用しているプラグインを追加して利用する。

設置場所はOn Singapore premiss server、またはクラウド上での設置を予定(ATEがクラウド運用を推奨している)

## 《Tracを利用するメリット》

- ・すでに各プロダクトで利用している公式インフラのため、運用実績がある
- ・構築ノウハウがあり、現在も保守が行われている
- ・開発内容説明書の自動生成等、ワークス用機能が充実している
- ・コミットしたものとの連携が取れる

## 1-3 開発開始～Trunk提出までの案件管理

## 《誰が使う》

Developer

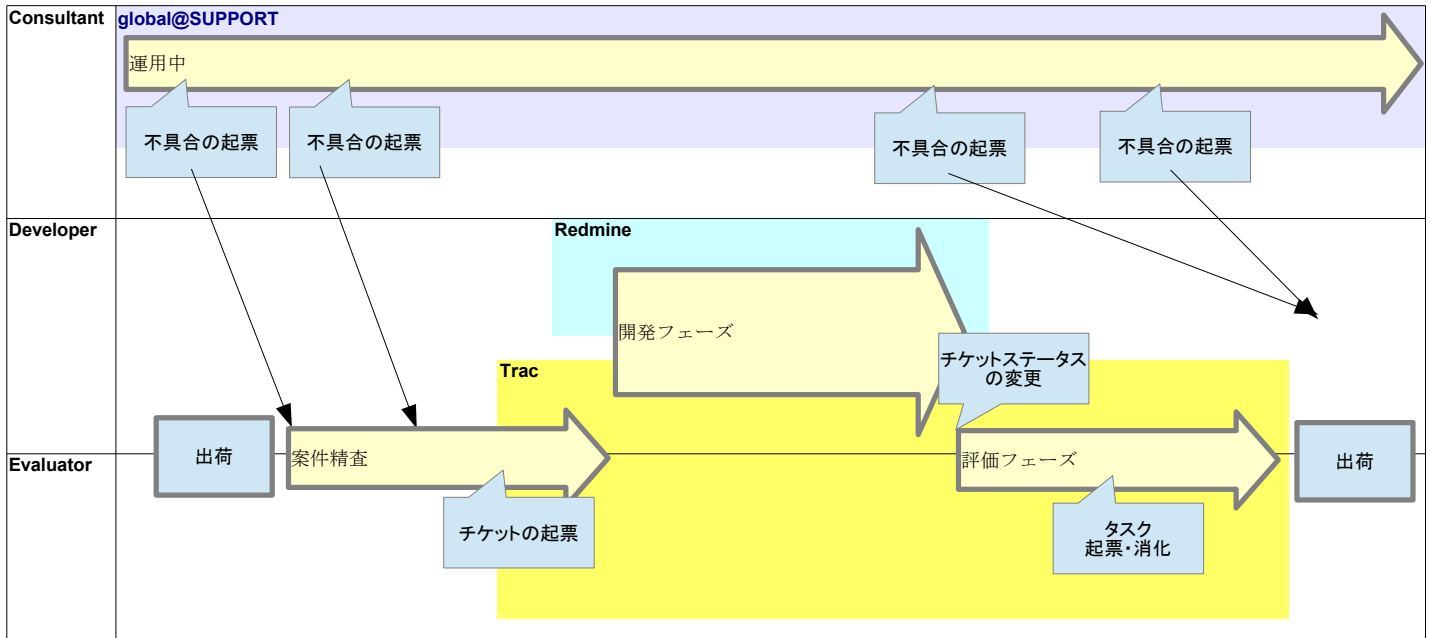
## 《場所》

「Shanghai Redmine」 <http://cjkshanghai-pc/redmine>

## 《Shanghai Redmineを利用するメリット》

- ・ 開発者間のコミュニケーションを容易に取れる
- ・ すでに開発間では利用しているため、導入コストがない

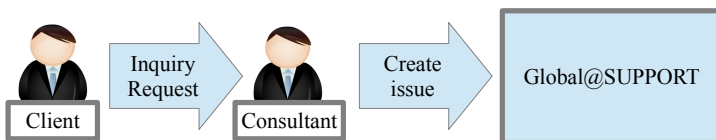
## 1-4 おおまかな流れ



## 2. 案件登録/不具合報告のフロー

## 2-1 起票について

お客様からくる案件を登録する場所はすべて「Global@SUPPORT」に統一する



## ＜不具合について＞

再現手順や詳細な現象を記載することによって、開発者は早期問題解決を行うことができる

## ＜開発案件について＞

従来はすべてのメリットを開発者が想定で考えていたが、よりお客様に近いコンサルタントがメリットを聞き出し、それを開発者に伝えていくことで、よりDevOpsとして業務を行っていくことができる

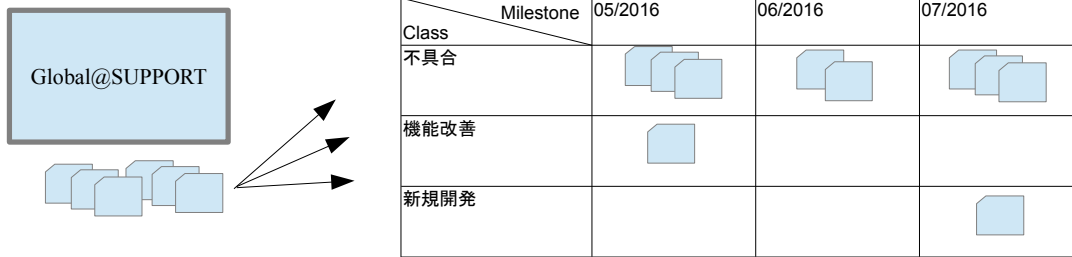
「メリットヒアリングシート」とは・・・

- ・ 業務内容
- ・ 業務目的マトリクス  
(誰が・誰に・いつ・目的・ツール・実施方法 × 現在の状況・現在の理由・理想の状況・理想の理由)
- ・ メリット内容

## 2-2 案件の精査

定期的に案件精査MTGを行う。

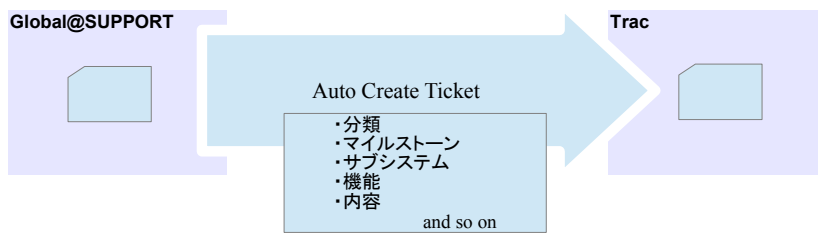
起票された案件の優先度、対応期限や難易度に応じて、対応する時期を検討することで開発計画を立てることを用意にすることができる



## 2-3 Tracへのチケット起票

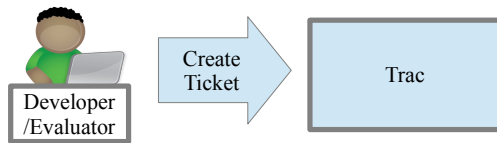
Global@SUPPORTから「Ticket作成」を行うと、自動でTracに対して新たなチケットが生成される

※ 現在はこのような機能はない。



## 2-3 開発内指摘について

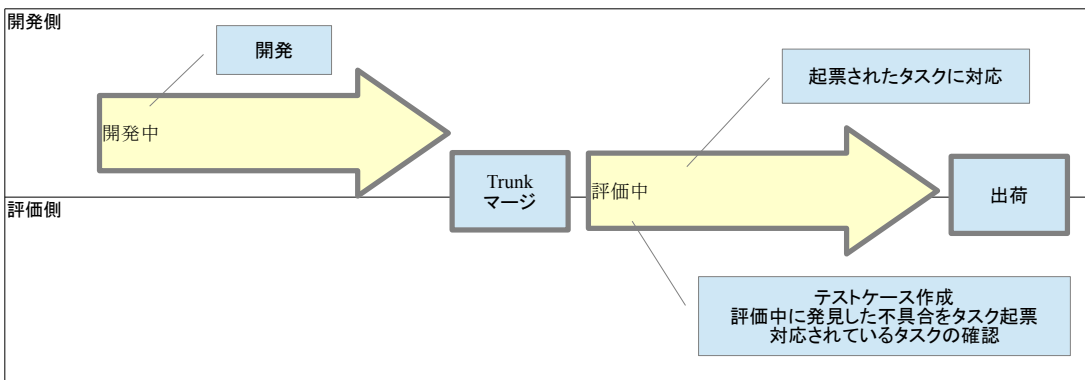
開発者・評価者が発見した不具合などの社内指摘は、直接Tracへチケット起票を行う



## 3. 案件の管理方法

### 3-0 おおまかな開発/評価フロー

下記のようなフローが想定され、これをチケットで進捗管理していく



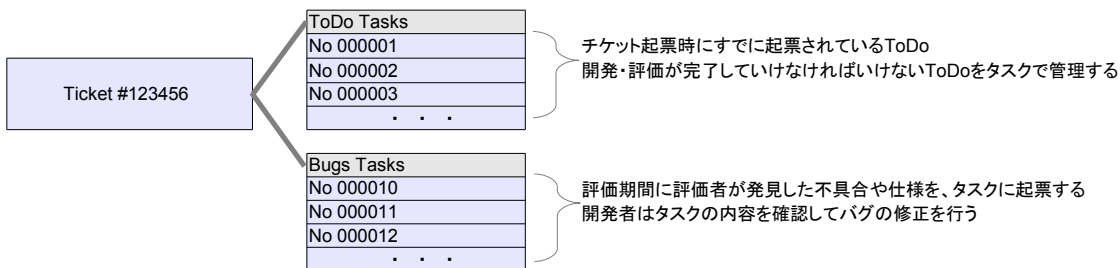
### 3-1 チケットでの案件進捗の管理

各チケットはステータスを持っているので、開発進捗、評価進捗にあわせてステータスを変更していく

開発フェーズ	ステータス	変える人	内容	この期間の管理を従来通り、Redmineで管理する
	not yet started	(Default)	チケット起票時のステータス	
	under development	Developer	開発がチケットの案件に着手した際 開発がステータスを「開発中」に変更する	
	merged trunk	Developer	開発が完了し、trunkへマージしたら 開発がステータスを「Trunk提出済み」に変更	
評価フェーズ	during the evaluation	Evaluator	評価が評価に着手した際 評価がステータスを「評価中」に変更する	
	completed the evaluation	Evaluator	評価が完了したら、「評価完了」にして チケットとしては出荷可能な状態になる すべてのタスクが「complete」している必要がある ※後述	

### 3-2 タスク管理

チケットに対して、完了しておかなければならないToDoや、評価から挙げられた指摘などを  
タスクとして管理することができる  
チケットにすべて関連付けられており、すべてのタスクが「完了」になっていなければ、チケット(案件)を完了にすることはできない



#### 3-2-A Todoの管理

開発/評価ともにやらなければならないことをToDoタスクとして管理し、タスクが完了すれば、ステータスを「完了」に変更する

<Regular>

#### 各種レビュー

不具合修正、新規開発で開催すべきレビューが異なるため、分類によって登録されるタスクがそれぞれ違う  
チケット起票時の分類に対して、自動でToDoタスクが登録される

<Regular>

#### Dropに対する確認内容

予定していたマイルストーンでの出荷が見込めず、止むを得なく次回に延期する場合  
進捗を変更したタイミングで、自動で確認すべきToDoが登録される

#### 提出物ごとの確認内容

例えば、cpny\_msginfo\_ins.datを出した場合、必ずcpny\_litinfo\_ins.datを出す必要があるなど  
関連するテーブルへのスクリプトも同時提出されているか  
(e.f.) cpny\_msginfo\_ins.datがコミットされた ⇒ ToDoタスクが自動登録

#### 3-2-B 評価中に発見された不具合の管理

評価フェーズ中の指摘はすべてタスク起票にて指摘を行う  
開発者はそのタスクで指摘されている内容を修正し、評価は修正された内容を更に確認し、タスクを完了にする

ステータス	内容
Development return	評価中に発見した不具合があった場合 不具合内容を記載してタスクを起票する
Response completed	開発がタスクに記載された不具合内容を修正し ソースをマージした際、開発はタスクステータスを、 「対応完了」に変更する
complete	評価は「対応完了」となっているタスクの不具合を 再度確認し、不具合が修正されていれば タスクステータスを「完了」にする

## 4. 今後の展望

### 4-1 開発内容説明書の作成

Tracには既に開発内容説明書(cdxml方式)の作成や、出荷物(モジュール、スクリプト類など)の記載を自動で行ってくれる機能が実装されている  
 開発者は、各項目を埋め、ソース・スクリプトコミット時にチケット番号をコミットログに記載するだけで自動で開発内容説明書を生成することが可能  
 また、今後グローバル側の出荷時はすべて英語の開発内容説明書を用意して、モジュールと一緒に出荷する

### 4-2 各種分析・集計

チケットには、機能名、画面名や不具合分類などを管理する項目があるため  
 どの画面の不具合が多いか、どのような不具合が多いかなど  
 QC活動を行っていく上での分析に利用することが可能になる  
 また、チケット数、タスク数、などを横断的に集計することが可能であり  
 出荷判定時のタスク数や、全体の進捗管理、などを行うことができる

### 4-3 バックアップ

Tracにある情報は過去のやりとりなどがすべて残っているため、例えばデグレードを起こした際の調査などで非常に重要な情報となる  
 また、予期せぬ事態が発生した場合に、すべてのデータが消えてしまうと業務を続けることができない  
 そのためにも、必ずバックアップを取る機構が必要  
 クラウド      ...      AMIを取得  
 オンプレミス      ...      Tracインストールフォルダ、データベースの退避

## 5. 課題

### 5-1 Global@SUPPORT

- ・ 英語化されていない画面・部分がある  
⇒ 修正が他部署
- ・ Global Payrollが非対応  
⇒ 新規プロダクトとして登録する必要がある

### 5-2 Trac

- ・ Gitとの連携の実績の少なさ  
⇒ Singapore ECでの利用実績がある
- ・ クラウド運用する場合のコスト  
⇒ 日本版Tracが\$181.09/month の費用が発生しているため、同等の費用がかかる見込み

Created : 2016/03/25

Author : Tanaka Yuki

Updated : 2016/03/31

## Purpose of this catalog

In current background, now global development process cannot achieve customer trouble management and process management.  
I want to form the way to issue customer trouble and the way to manage process from development to shipment.  
And, in nearly future, I want to decide more detailed development process that be based on this proposal.

## Current problem

There is no location where the found bugs and request new feature is managed on.  
Their cause following problem.

- Consultant contacts developer directly, so this communications don't remain.
- The priority is decided by personal efforts
- We cannot know all issue problem

And, in development process, there are many problems.

- Process management is loose
- The ToDo that must be finished before shipment is not managed
- The shipped modules, scripts are not managed

## Summary

1. Location
2. Registering trouble/Process of issue
3. The way to manage process of development
4. Future view
5. Problem

## 1. Location

**1-1 the Location where we register customer trouble****«Requirement»**

it can register customer trouble with detail information

**«Who use it»**

Consultant

**«Location»**

「Global@SUPPORT」 <https://support.worksap.com/>

**«Advantage»**

- the consultant can customer trouble in the same location where consultant communicate customers
- it is easy to research issue or trouble (sort by customer, priority, deadline,,)

**1-2 the Location where we managed trouble and task progress****«Requirement»**

it can managed task progress and the information that

**«Who use it»**

Developer & Evaluator

**«Location»**

「Provisional Name: Trac for CPNYGP」 (This does not exist yet)

We will use the "PRM" ([http://sho-gouki/trac/trac\\_dev/wiki](http://sho-gouki/trac/trac_dev/wiki)) that is made by ATE and based on Trac,  
and install plugin for CJK that is made by CJK

I plan that it will be set on Singapore premiss server or on cloud  
(ATE recommend cloud)

**«Advantage»**

- Many division use it, and it is official infrastructure of WAP. So It has results
- Many people have a knowledge of constructing PRM, and now ATE maintain it.
- It have many function for WAP (e.f. Auto document creator)
- it can connect with the thing committed by developer (e.f. source code, scripts)

### 1-3 Progress management while Starting development and merging Trunk

#### «Who use it»

Developer

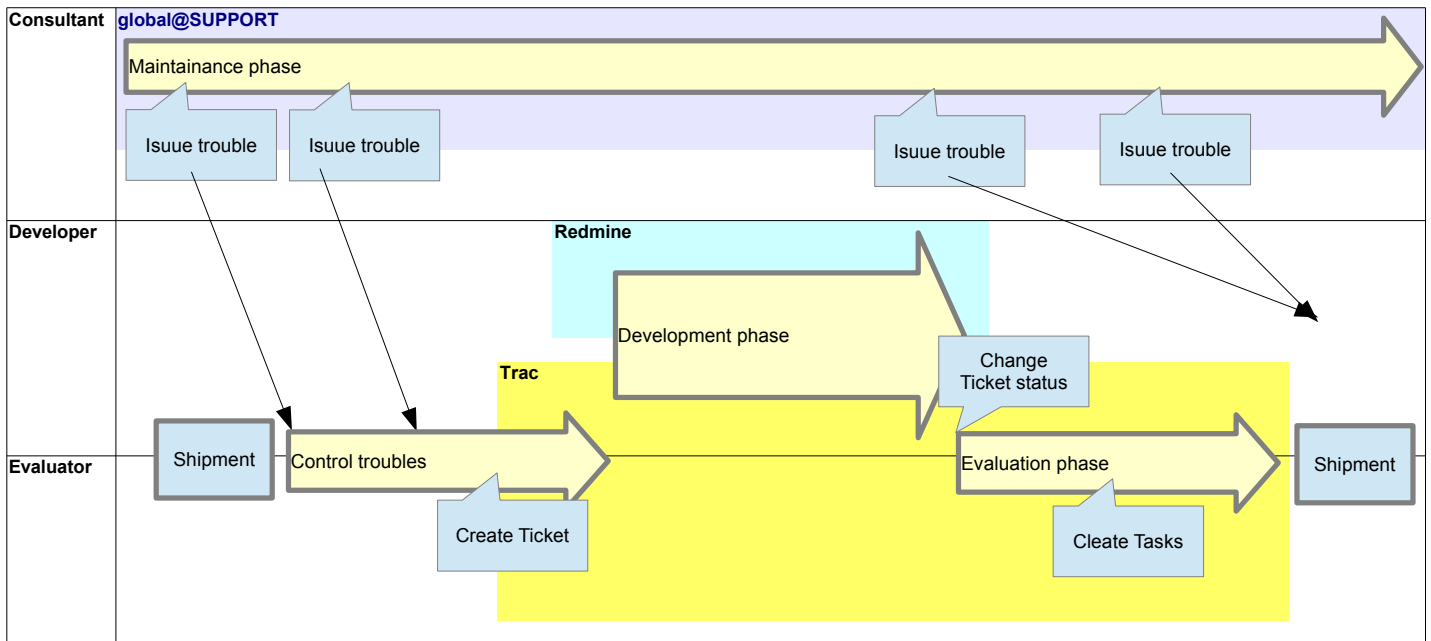
#### «Location»

「Shanghai Redmine」 <http://cjkshanghai-pc/redmine>

#### «Advantage»

- It is easy for developers to communicate each other.
- Singapore member have used it already, so we need not to spend anymore cost for introductio

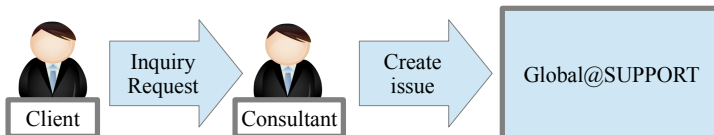
### 1-4 About process



## 2. Registering trouble/Process of issue

### 2-1 about issue trouble

All client trouble will be registered on "Global@SUPPORT"



#### <About Bugs>

The consultant can register trouble with procedure for reproducing of the bugs and detail bug information on [Global@SUPPORT](#). So developer can find cause more quickly by this information.

#### <About new feature, improvement function>

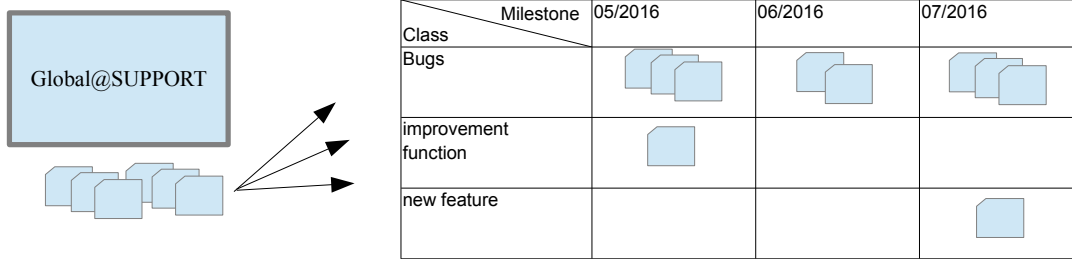
All developers think all of customers benefit by only their supposition in conventional way. Consultant is always contact with customer and they write "Merit Hearing Sheet", This is way DevOps will be more able to develop and operate for customers

「Merit Hearing Sheet」 is...

- contents of customer business
- Matrix of business target  
(Who/Whom/When/Portuse/Tool/How to × Current situation/Current cause)
- benefit for this customer

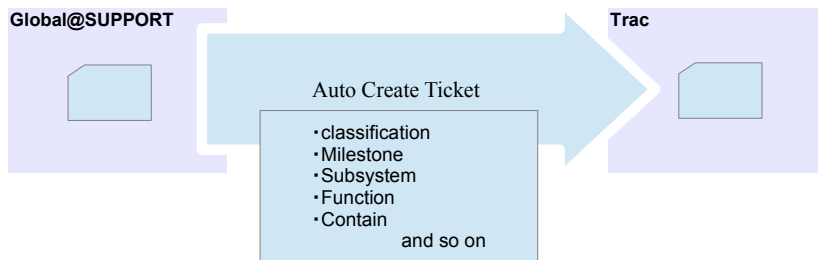
## 2-2 Control troubles

We will have a "Control troubles meeting" regularly.  
In order to plan development schedule, we decide the Milestone and divide into classes  
by priority, deadline, difficulty of issued troubles



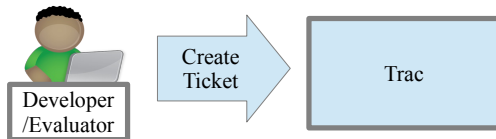
## 2-3 Create Ticket for Trac

Global@SUPPORT can create Trac's Ticket automatically when the user push "create ticket for Trac" on Global@SUPPORT  
\* now there is no such function



## 2-3 Issue trouble in company

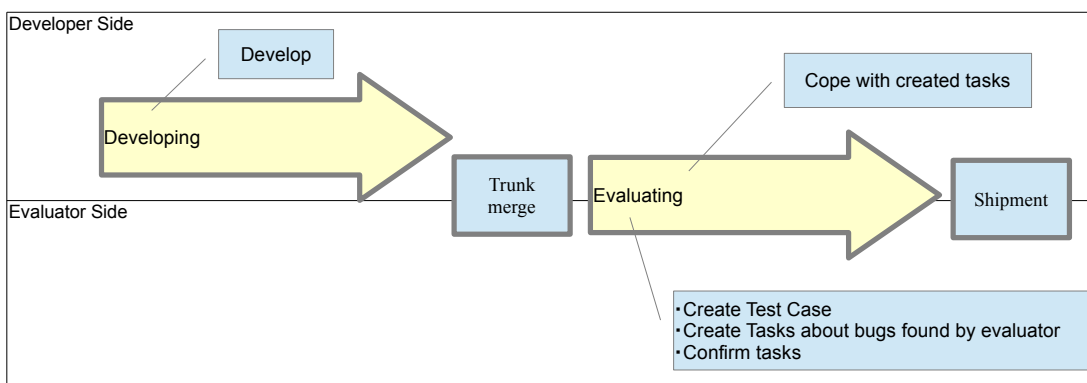
If developer or evaluator found bugs, they create Trac's Ticket themselves.



## 3. The way to manage process of development

### 3-0 About Development/Evaluate process

Following process is supposed, so we will manage this process by Trac





### 3-1 Manage progress by Ticket

Ticket have status, we will change status if there is progress

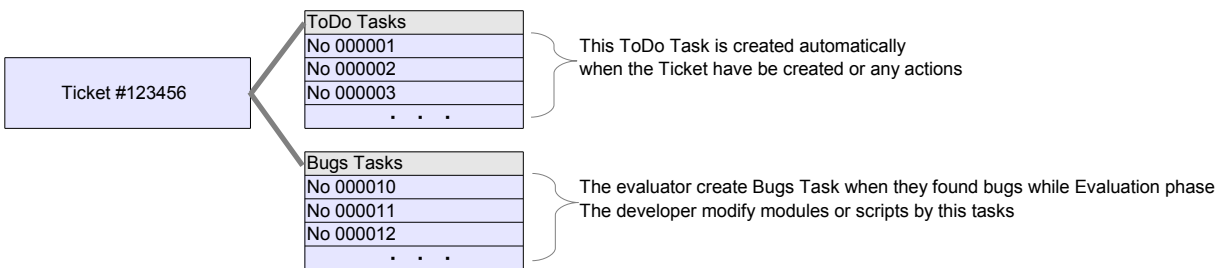
Status	Who changes	When/Content
not yet started	-	Default status
under development	Developer	When the developer start develop, developer change status
merged trunk	Developer	When the developer have finished and commit to trunk developer change status
during the evaluation	Evaluator	When the evaluator start evaluate, evaluator change status
completed the evaluation	Evaluator	When the evaluator have finishd evaluating evaluator change status It is necessary that all tasks status is completed

in addition manage progress by Redmine

### 3-2 Manage progress by Task

Trac can manage that ToDo that need to be finished and the bugs reported by evaluator as "Task"

All tasks is related Ticket and It is necessary to finish Ticket that All tasks status is completed.



### 3-2-A Management Todo

Developer and Evaluator's ToDo is managed as "ToDo Task", and when they have finished this task, they change task status "completed"

<Regular>

#### Review

The reviews that must be had are difference between "Bug fix" and "New feature", also registered ToDo tasks are different each ticket class  
When the ticket are created, the ToDo tasks is registered automatically

<Irregular>

#### When the ticket is dropped

If we cannot ship by milestone and postpone shipping,  
when progress is changed, the ToDo that must be confirmed is resistered automatically .

#### Confirming each committed things

For example, when developers commit "cpny\_msginfo\_ins.dat", developers must confirm that have they commit other scripts for rerated tables.  
also, the ToDo tasks is registered automatically by committed things.  
(e.f.) cpny\_msginfo\_ins.dat in committed ⇒ the ToDo task "you must add record to "cpny\_litrinfo\_ins.dat" is registered.

### 3-2-B Management bugs report

If evaluator find bugs in evaluation phase, evaluator report bugs as "Bugs task"

Developer modify the modules or the scripts by this tasks, and evaluator confirm this modification.

If this modification has no problem, evaluator changes task status.

ステータス	内容
Deveropment return	If evaluator finds bugs evaluator creates task as reporting bugs
Response completed	When developer modify modules or scripts and commit to official branch, developer changes task status.
complete	Evaluator confirms tasks that status is "Response completed" and there is no problem, Evaluator changes status

## 4. Future view

### 4-1 Create document for shipment

Trac have a functions that are creating document for shipment, recording committed things automatically.  
 If developer input each points and commit things with ticket number that is written in commit log,  
 Trac can create document for shipment automatically.  
 And we must create document for shipment by English when "LPP" is released officially, so we must decide document format

### 4-2 Analysis/Totalization

Ticket has many items recording function name, screen name, classification of bugs and so on  
 Also their information is very useful for QC action, for example which screen does occur most numerous bugs, what kind of bugs is occurred.  
 And, Trac can totalize number of ticket, tasks, and classification of developer, evaluator, and any items,  
 So, we can make document of ship judgement and manage whole progress

### 4-3 Back up

Trac stores all information, so this information will be very important to research degrade.  
 And, If this informations will be lost, we cannot continue our business.  
 In order to avoid it, we must make back up.  
 in case of cloud PC           ...    get AMI  
 in case of PC on premiss   ...    copy folder of installed Trac and dump of Trac DB

## 5. Problem

### 5-1 [Global@SUPPORT](#)

- Many screens or words is Japanese still.  
 ⇒ We cannot modify it, so this is managed by other department
- Global Payroll is unmatched  
 ⇒ We must resister it as new product

### 5-2 Trac

- There is a few experience of Git connection  
 ⇒ Singapore EC use Trac with Git now
- Cloud cost  
 ⇒ In Japan, cloud for Trac spend about \$181.09 monthly I estate some cost in Singapore if we use cloud