# RT/CSCF/MFCF

Reference: <https://cs.uwaterloo.ca/twiki/view/CF/STToRTInvestigationFactors>

## Next meeting

1. RT4 overview and training
   1. UI
   2. Dev machine
2. QA
3. Plan next meeting to discuss
   1. critical features and nice to haves (see ‘Critical Factors and Features’ below)
   2. Ticket roles (see ‘Ticket roles’ notes from Robyn below)

## Mon Jun 6 2016

### Attendees

Lawrence E Folland <lawrence.folland@uwaterloo.ca>; Fraser Gunn <fraser.gunn@uwaterloo.ca>; Daniel Allen <drallen@uwaterloo.ca>; Robyn B Landers <rblanders@uwaterloo.ca>; Jim Johnston [jim.johnston@uwaterloo.ca](mailto:jim.johnston@uwaterloo.ca); Lisa Tomalty <ltomalty@uwaterloo.ca>

### Action items

1. Lisa to schedule training meeting with all incl Jeff
2. Lisa to request dev server for cscf and mfcf and set up perm for all on mfcf-teest queue

### Notes

1. Timing
   1. Considering May 2017
   2. Hope for CSCF/MFCF to move at same time
2. Training
   1. By request (Lisa and/or Jennifer)
3. Getting familiar with IST’s current RT
4. Permissions discussion
5. Possible use of owner/admin cc
6. Linking tickets

## Ticket Roles

Often for simpler requests the responsible person is the same as the owner.

When they are different, the difference is that the responsible person is where the buck stops -- making sure the work happens, but not actually doing the work. The work is done by the owner(s).

The remark about personal contact is a CSCFism. In MFCF it's more about who's in charge of some project (e.g. with multiple sub-requests), or who's responsible for some service, while somebody else does the actual work for whatever the particular matter at hand is.

Lawrence gave the example of Gord's being the point of contact for the PLG group (or whatever) so he's responsible, even if someone else happens to be handling the task. I gave the example of my being the security guy in MFCF, so I'm responsible even if someone else is fixing whatever security issue has popped up.

Robyn

---------

Responsible:

The "Responsible" field in a work request identifies the single entity that is responsible for resolving the request. It is either a UW userid, or an e-mail address of another work tracking system, typically "request@ist".

This is potentially distinct from the owners of a request, who are responsible for doing the actual work. It is usually the person who the requesters of the work have personal contact with.

The default for the "Responsible" field is the first name in the list of owners that has permission to modify requests, if any of them do.

Otherwise it's just the first name in the list.

------------

Owner:

The "Owner" field in a work request identifies who (or what) is doing the work to resolve the request. An owner is either a UW userid, or an e-mail address of another work tracking system, typically "request@ist".

This is distinct from who/whatever is responsible for the resolution of that work; that's recorded in the responsible field.

When there are multiple owners, the extra names in the list identify those who may be advising, or in fact doing part of the work best suited to their skills and availability. Regardless of their contribution, it's the first person/system in the list who is currently doing the work. That name may change over time.

--------------

AdminCC:

The "Admin CC" field in a service record identifies who wishes to monitor the work, by receiving the same generated email that the owner does, and by having the same access to the record that the owner does. This is typically someone in a supervisory or management role.

## Critical Factors and Features

• We need a smooth process for making improvements to the software, whether it's in order to add or modify features, or in order to accommodate our workflow (including changes we make to our workflow to meet our client needs).

• We need prompt updates about planned changes to the software that are outside of our control.

• Feature: tracking time spent, with a convenient interface (ie, not editing the total time worked)

• Feature: ability to view all tickets in queue, not just ones you're involved with

• Feature: Item dependencies and see-also

• Feature: Ability to update tickets via email

• Feature: Attachments to related documentation- uploading files, as well as links to private/public files.

• Feature: Auto lookup barcodes in inventory and hostnames in inventory and DNS, fill in corresponding info and room number, warn on typos, canonicalize hostnames, link barcodes and hostnames to inventory

• Feature: Auto lookup users and email addresses, canonicalize, show real name, link to WatIAM, warn on typos

• Feature: Handle simultaneous update by multiple users of multiple fields in the same request without locking or losing updates

• Feature: Sort search results by due date, date created, or last acted (or any field displayed)

• Feature: Arbitrary SQL searches (e.g. date\_created>unix\_timestamp()-3600\*24\*7 and subject like '%print%') with ability to bookmark and API (e.g. needed for Bill's run-on sentence)

Important Factors and Features

• We'd like to collaborate with IST and other RT stakeholders to implement an RT-connected Knowledge Base

• Feature: batch dependency creation with particular fields set to arbitrary values

• Feature: flexible search; if possible, ability to include SQL (eg., Fraser will fine-tune search query with SQL wildcards such as "evergreen%MC%3007")

• Feature: API so our inventory system can continue to report on related work items

• Feature: custom field, for subscription code, that we can easily extract work-hours on particular codes, and replicate the ST functionality within the "subscriptions" database.

• Feature: recording data to allow budgeting / managing spending using cost-estimate (and keywords and dependency trees - such as summing cost-estimates including dependencies)

• Feature: direct access to the uploaded attachments, eg., smbmount

• Feature: Arbitrary choice of columns and their order and optional abbreviation in search results with ability to bookmark and API

## Nice-to-have Factors and Features

Feature: command-line access to perform searches and updates