# [Agile Ceremonies](https://confluence.schroders.com/display/GCFN/Agile+Ceremonies)

|  |  |
| --- | --- |
| Stand-up | Daily 10:00-10:30, **5.C.3** |
| Backlog Refinement I | Every Friday, from 10:30-11:00, **SKYPE** |
| Backlog Refinement II | Dynamic, last Monday/ Tuesday of each sprint, **SKYPE** |
| Sprint Planning | Last Tuesday of each sprint, from 10.30 to 11.30, **SKYPE** |
| Demo | Last Monday of each sprint, from 17.00 (may be starting right after team meeting), **TBA** |
| Retrospective | Last Tuesday of each sprint, from 16:30 to 17:00, **TBA** |

House Rules

In below, the house rules are listed which we agreed to follow during our SCRUM ceremonies

**Stand-Up Meeting**

The stand-up meeting is an information exchange meeting

* The duration of the meeting should not exceed 15 minutes
* Physical attendance for all Luxembourg team members is preferred
* TFS will be used to guide the meeting
* Every team member should attend physically, if possible
* Each member should prepare his 30 seconds of fame during the meeting
* Typical content: what have I done, what I plan to do, do I have an impediment?

Consider Preparation:

* Log the key achievements of the day so that credit can be taken from them
* TFS statues of owned work items and tasks are up to date

Brief discussions and clarification questions are allowed, but in depth discussions and analytical activities should be postponed after a meeting to the interested audience.

**Backlog Refinement I**

*(30 mins)*

* Backlog items can be approved during the refinement meeting.
* The order of backlog items is reviewed and cross checked towards the PI plan

Attendees:

* Business team is mandatory
* Developers are expected to attend, can however be excused for good reason (sprint goal under risk)

**Backlog Refinement II**

*(30 mins)*

* Processing of backlog items top/ bottom

Attendees:

* Developers are mandatory
* Business is on stand-by: not necessarily connected, by available for Skype calls/ chats

**Sprint Planning**

* The sprint plan should contribute to the completion of the [Product Increment Planning](https://confluence.schroders.com/display/GCFN/Product+Increment+Planning).
* The sprint capacity for each individual must be updated (holidays, sickness etc.)

For business:

* Prior meeting: kindly ensure that the PBI sequence is top/ bottom
* Prior meeting: approve items and/or add missing content
* **Attendance to the meeting is optional**, but you should be reachable through Skype

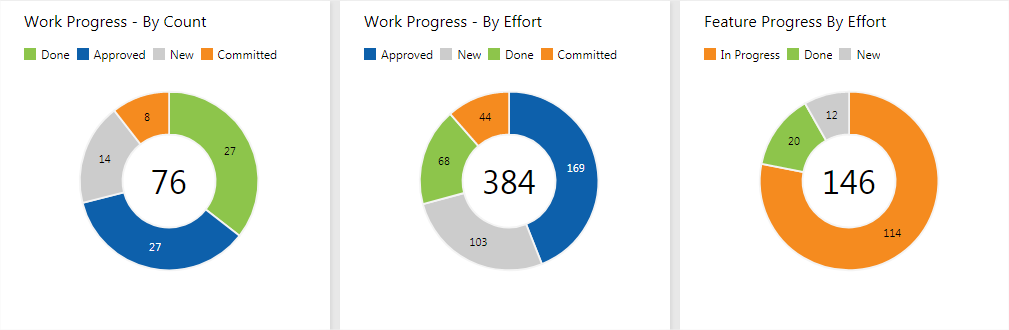
For developers:

* Attendance required on Skype
* Ensure that all items (top-bottom) have tasks assigned -including 'Remaining Hours' (=equals total hours)
* If an item has already tasks, cross-check whether the tasks are up to date and lead to the completion of the item
* Drag the item under planning into the sprint
* Fill the assigned to field
* Change status to ‘committed’ **check:**Is parent feature 'In Progress'? If not update status of parent

**Sprint Demo**

* All work items with value-area business which have not been opt-ed out will be presented
* Work items with value-area ‘architecture’ are presented after all functional work items (business representatives are not required for this part of the meeting)
* Work items have to be demonstrated from the **DEV** or **SIT** environment where applicable
* The presentation order is by owner of the work item (top-bottom)
* A page in this confluence space needs to be created with the content of the sprint demo. Within this new page, a presentation will need to be created (better to use the previous version as sample) in order to list what was on the plan, what has been done and then what will be demo (information used for this come from TFS)
* After demo we should review % of Completion of the PI objectives with the business and update EPMO (Dedicated page created for EPMO updates [EPMO updates - Sprint percentage of completion](https://confluence.schroders.com/display/LUXTA/EPMO+updates+-+Sprint+percentage+of+completion))

**Sprint Retrospective**

* A page in this confluence space needs to be created with the content of the demo, under [Retrospectives](https://confluence.schroders.com/display/GCFN/Retrospectives).
* Include the Burn Down chart
* Include the Product Increment Progress Pies  
  

# [Definition of Ready](https://confluence.schroders.com/display/GCFN/Definition+of+Ready)

## Product Backlog Item (PBI, User Story)

A product backlog item is a work item which is typically assigned to a team member and has to be completed within the sprint.

* The item contains enough level of detail to be implemented (analysis performed)
* The item contains the necessary tasks in order to complete the item
* The item can be done in a single sprint
* The item's acceptance criteria are clear and measurable by the user and tester (if not deeply technical)
* The item is not a duplicate (the functionality is not overlapping with another PBI)
* The links of the item are meaningful (parent and related items belong to the same Epic/ functional context)
* The item would be demo-able when the work item is delivere
* Confluence documentation exists or is updated by the PBI
* **The item is in status 'approved' by the capability owner or the deputy  this step implies that above criteria are met**

## Bug (Defect, Malfunction)

A defect is a work item which is may contradict natural expectation or a specified behavior.

* ~~The steps-to-reproduce contain a GIVEN-WHEN-THEN sequence of steps which lead to the misbehavior~~
* The steps-to-reproduce contain the necessary parameters to repeat the misbehavior
* The bug is linked to a work item where the specified behavior is described (this is important otherwise the backlog may not show the bug correctly). Exceptionally a generic work item may be referrenced, if the specification is described in Confluence (in this case the link must be part)
* The item is not a duplicate (the functionality is not overlapping with another PBI)
* The item is in status 'approved' by the capability owner or the deputy

# [Definitions Of Done](https://confluence.schroders.com/display/GCFN/Definitions+Of+Done)

This article defines several definitions which apply to the sprints and product increments (not-agreed check points are ~~strikethrough~~)

* [Definition of Done: Product Backlog Item (PBI, User Story)](https://confluence.schroders.com/display/GCFN/Definitions+Of+Done#DefinitionsOfDone-DefinitionofDone:ProductBacklogItem(PBI,UserStory))
* [Definition of Done: Bug](https://confluence.schroders.com/display/GCFN/Definitions+Of+Done#DefinitionsOfDone-DefinitionofDone:Bug)
* [Definition of Done: Impediment](https://confluence.schroders.com/display/GCFN/Definitions+Of+Done#DefinitionsOfDone-DefinitionofDone:Impediment)
* [Definition of Done: Sprint](https://confluence.schroders.com/display/GCFN/Definitions+Of+Done#DefinitionsOfDone-DefinitionofDone:Sprint)
* [Definition of Done: Feature](https://confluence.schroders.com/display/GCFN/Definitions+Of+Done#DefinitionsOfDone-DefinitionofDone:Feature)
* [Definition of Done: Epic](https://confluence.schroders.com/display/GCFN/Definitions+Of+Done#DefinitionsOfDone-DefinitionofDone:Epic)
* [Definition of Done: Product Increment](https://confluence.schroders.com/display/GCFN/Definitions+Of+Done#DefinitionsOfDone-DefinitionofDone:ProductIncrement)

Definition of Done: Product Backlog Item (PBI, User Story)

A product backlog item is a work item which is typically assigned to a team member and has to be completed within the sprint.

* All tasks of the PBI are 'done'
* ~~Unit tests are covering at least 60% of the described functionality (aim towards 90%)~~
* ~~No warnings as error-remain (implies: no warnings are excluded by developer)~~
* ~~Coding guidelines are followed~~
* Code review has been passed
* The code is in the **master** branch, any dedicated branch is deleted
* ~~The functionality has been demonstrated to business~~
* The functionality is available in SIT and UAT
* ~~Confluence documentation exists and is updated with the latest changes (caused by the PBI)~~

Definition of Done: Bug

The following applies to software components which are developed internally:

* The steps-to-reproduce are updated/ maintained in a TFS bug
* ~~The TFS  bug~~*~~history~~*~~contains all relevant information about the information exchange with the third party~~
* ~~The root cause of the defect has been identified~~
* The bug fix is described functionally right below the GWT (Gherkin Language)
* ~~A unit test has been written to demonstrate the failure (if applicable -yes by default)~~
* The patch has been implemented
* The patch has been released for testing in SIT and UAT
* The patch has been forward-ported, if applicable
* ~~Testing has been completed (either by QA or the owner of the defect -not the developer)~~

For **external** defects, the definition of done is obviously simpler:

* The steps-to-reproduce are updated/ maintained in a TFS bug
* ~~The~~*~~history~~*~~contains all relevant information about the information exchange with the third party~~
* The patched version of the third party software

Definition of Done: Impediment

* ~~The impediment has been communicated to the stake holder(s)~~
* ~~The impediment has been removed - or - mitigation PBIs have been created.~~

Definition of Done: Sprint

* All assigned work-items have been 'done'
* ~~There is no bug related to a work item active~~
* ~~All items are demonstrated to users~~
* Sprint retrospective has been completed
* Sprint demo has been completed \*(if not strictly IT related)
* EPMO website has been updated with the progress

Definition of Done: Feature

The feature represents a functionality on a high level as expressed by the user. Typically a feature will span several sprints, and can even overlap product increments.A feature is done:

* All the Acceptance Criteria are met and confirmed by the user(s)
* All testing activities have been completed and no critical bugs remain
* All child-backlog items are 'done'

Definition of Done: Epic

* ~~All the Acceptance Criteria are met and confirmed by the user(s)~~
* ~~All testing activities have been completed and no critical bugs remain~~
* All its features are 'done'

Definition of Done: Product Increment

* All objectives have been met or some objectives have been met and others are escalated timely
* EPMO website has been updated with the progress

# [TFS Patterns and Practices](https://confluence.schroders.com/display/GCFN/TFS+Patterns+and+Practices)

Common Tags

| **TAG** | **DEFINITION** | **USE ON** |
| --- | --- | --- |
| **PI OBJECTIVE** | Marks the feature/ PBI | Epic,Feature |
| **~~EXTERNAL~~** | Indicates that the item belongs to someone outside of the SCRUM team | Feature, Product Backlog Item, Task |
| **IN TESTING** | Indicates that the feature is under testing (please note: backlog items have a testing activity) | Feature |
|  |  |  |

Roles

|  |  |
| --- | --- |
| **Capability Owner/**  **Analyst** | The person who is writing the Product Backlog Item, typically an analyst. |
| **Reviewer** | The person who reviews the item for content, consistency and consistency |
| **Implementer** | The person who is responsible that the work item is completed (or leads others towards completion). A software development item is typically performed by a software developer; an analytical item is performed by the analyst (common sense). |

Please note that individuals can have more than one role within the team!

Life Cycles

In order to address an outstanding IT Risk/Audit gap, three **new** states and new functionality have been introduced into TFS for user stories (Product Backlog Items or PBI’s) in-line with VSTS.

*New States:*

* **READY**
* **IN PROGRESS**
* **REVIEW**

*How New States appear in the TFS Workflow:*

*https://confluence.schroders.com/download/attachments/29776110/image2018-10-24_14-33-4.png?version=1&modificationDate=1540384384470&api=v2*

| **State** | **Accountable** | **Mand/Opt** | **Definition** | **Notes** |
| --- | --- | --- | --- | --- |
| **New** | Any | Default | Any new Story | No Change |
| **Approved** | CO | Optional | CO can use this state to differentiate stories that have been prioritised for further refinement | No Change Approved user and date fields are auto-populated by TFS user performing the action |
| **Ready** | CO/Team | Audited | CO agrees that the story reflects their expected outcomes and Team agree that it meets their "Definition of Ready" for sprint planning | New Audit Step - If CO delegates this TFS action, a note should be added to that effect in TFS |
| **Committed** | Team | Audited | Team has understood the story, estimated and committed to completing the work in the sprint | No Change |
| **In Progress** | Team | Optional | At least one of the Team members has started on the story |  |
| **Review** | Team | Optional | Team has agreed that the work for this story has met their "Definition of Done" for the CO to review | This is used now for indicating testing |
| **Done** | CO | Audited | CO agrees that work has met their expectations, Acceptance Criteria and the team's "Definition of Done" | No Change  Audit requirement - CO has signed off story - If CO delegates this TFS action, a note should be added to that effect in TFS Closed user and date fields are auto-populated by TFS user performing the action |

**What does this mean to you?**

**As a Capability Owner:**

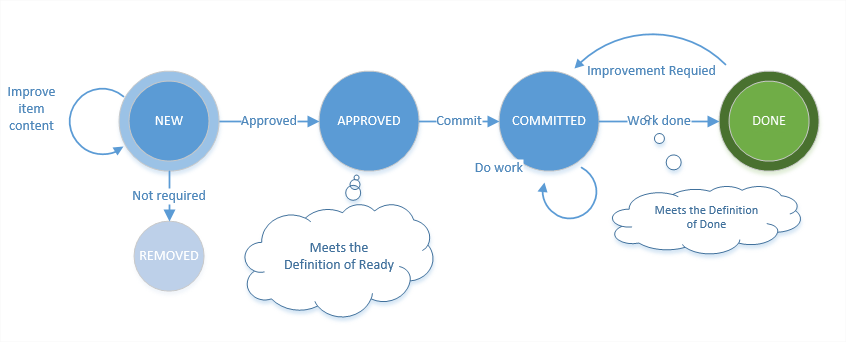
* You are still responsible for any story moving to Approved (if you choose to use this)
* You are responsible for any story moving to Ready. This means:
  + It accurately reflects the functionality and behaviour you anticipate
  + It has been through a minimum level of refinement for the team to question and clarify understanding
  + It meets the team’s “Definition of Ready” – a good example is [here](https://confluence.schroders.com/pages/viewpage.action?spaceKey=CC&title=Definition+of+Ready)
  + You are still responsible for any story moving to Done, but this must be with the agreement of the team meeting their “Definition of Done”

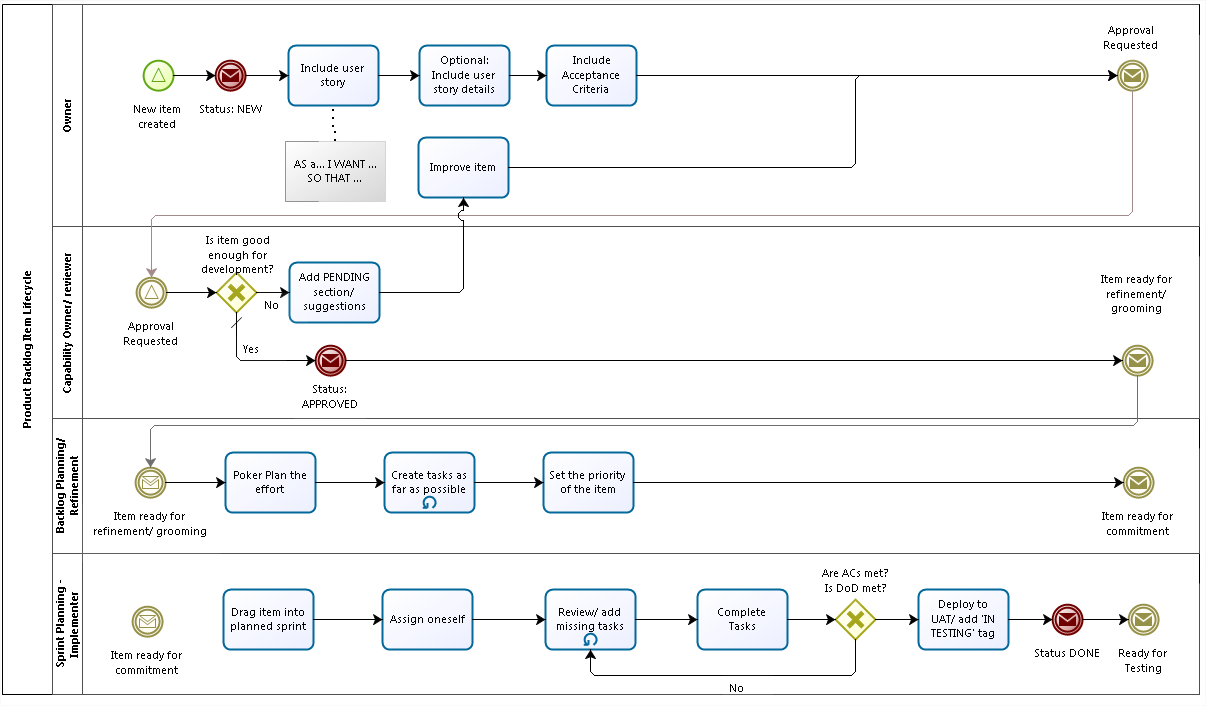
Where the CO delegates this action to one of the team, the CO will still remain accountable for the transition and a note should be added to the story to reflect the CO’s input and approval.  Any delegation should be clearly agreed within the teams “Definition of Ready” and “Definition of Done”.

**As a Team Member:**

* You are still collectively responsible for any story moving to Committed as part of the sprint planning
* You are responsible for any story moving to In Progress (If you choose to use this)
* You are responsible for any story moving to Review (If you or the CO choose to use this)

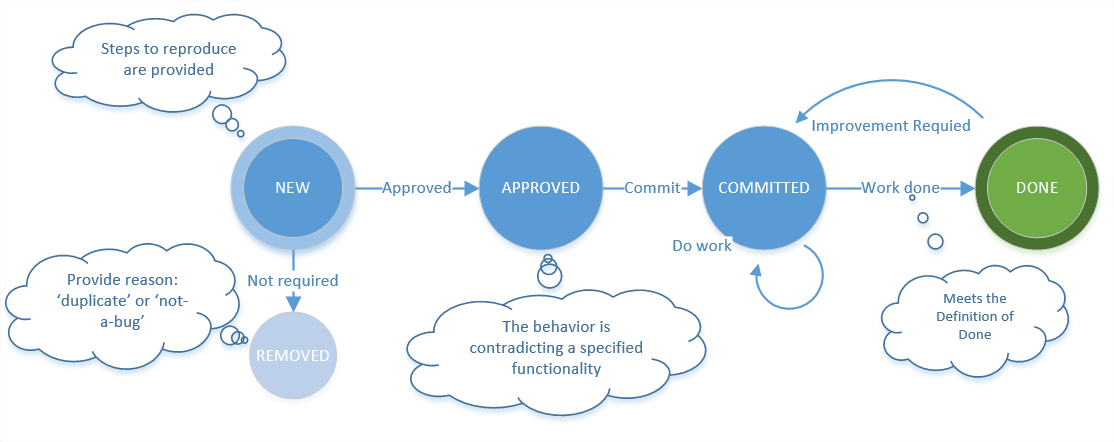
**Product Backlog Item**





Please note that the lifetime of a product backlog item should not exceed a sprint.

**Defect**

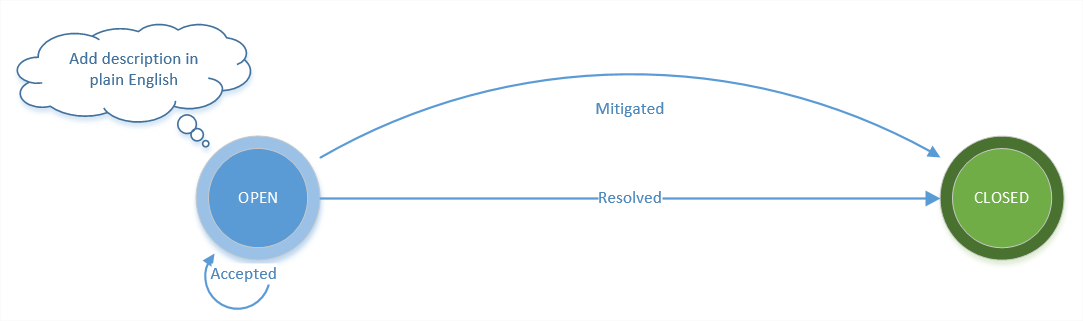


Please note that the lifetime of a defect backlog item should not exceed a sprint.

**PI Objective**

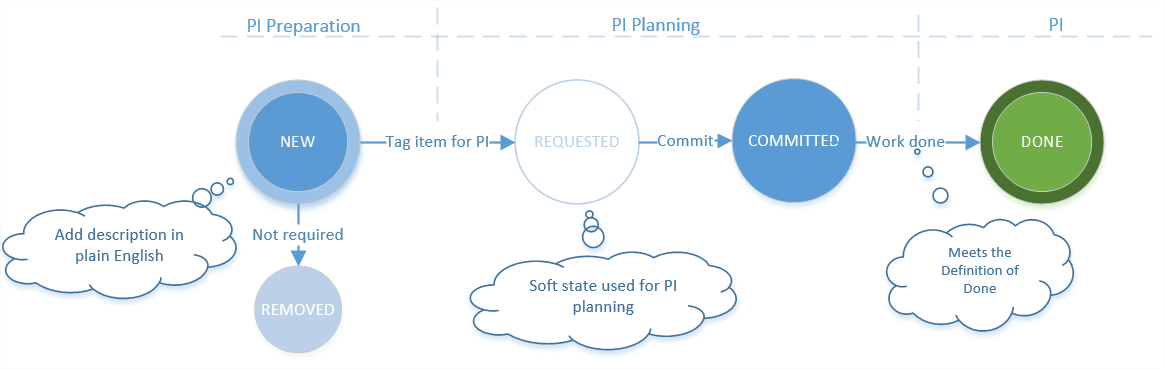
PI Objectives are work items, which are defined, tuned and agreed during the PI planning. One PI objective are be related to one or multiple features

**Impediment and Risk**



**Feature**

A featue is a (small) functional unit which should be completed within a Product Increment, except when it is strecht or descoped. Each feature needs to have a business value and an effort estimate. Features need to be linked (parent/ child) to epics.



**Epic**

An epic is an umbrella for several features to fullfil a strategic goal. Epics are not necessarily completed within a PI. Epics should be related to 'portfolio epics'.

