

Welcome to MkDocs

For full documentation visit [mkdocs.org](https://www.mkdocs.org).

Commands

- `mkdocs new [dir-name]` - Create a new project.
- `mkdocs serve` - Start the live-reloading docs server.
- `mkdocs build` - Build the documentation site.
- `mkdocs -h` - Print help message and exit.

Project layout

```
mkdocs.yml    # The configuration file.
docs/
  index.md    # The documentation homepage.
  ...        # Other markdown pages, images and other files.
  ./agile/work-item-states.md
```

Last update: May 1, 2022

Daily Standup

Daily Standup's are Team meetings. These meetings are designed to quickly report status and for the team to hold one another accountable to updates regarding work the team committed to in pursuit of the current Sprint goal. This is a meeting for the team to hold one another accountable and to assist with achieving the Sprint goal. Stick to the meetings intent.

Three Questions

Team members **MUST** proactively answer the following 3 questions only:

1. What did I work on yesterday?
2. What do I plan on working on today?
3. What are my impediments (if any)?

(pick the next team member to give their update)

Guidelines

Here are some other guidelines to hold one another accountable to staying on target:

- **Be prepared with your updates** the 3 questions and give your update when you are ready (no assigned order)
- **Developer dashboards may be for visibility/ context, not to answer the 3 questions above,** and mention ticket #'s as needed
- **Use the phrase " ... and other stuff you"** worked on tasks unrelated Sprint (like an interview, a task unrelated to the Sprint from leadership, etc.).
- **Say " Enough, Let's Move On."** if someone rambles or strays to indicate to all of us
- **Say " impediments" , not " blockers"** indicate your progress is impeded and you need help from the Team or others to progress on the Sprint Goal
- **Do not elaborate on impediments** in the Daily Standup, the Scrum Master will track and they will be addressed in the Daily Sitdown or separate meetings - or feel free to reach out to needed Team members as needed.

Work Item Prioritization

Order of priority in determining which items to work on (and try to complete) first:

1. Hybrid Production Hotfix items (most recent release)
2. Hybrid current release User Stories and Defects
3. Hybrid current release production Bugs (Priority 1-3 only)
4. Hybrid next release User Stories (typically the next month)
5. Hybrid next release Bugs (typically the next month)
6. Web X Emergent Work
7. Priority 4 Bugs

If you have multiple items within the same priority tier, refer to the Severity or Priority on the work item. As usual, if you are still uncertain, you should ask Justin, Seth, myself or Matt.

Last update: May 1, 2022

Work Item States

These diagrams are enabled in mkdocs using [the Mermaid Plugin](#)

Work Items are managed by state in order to ensure **priority is set**, requirements are gathered, and progress is tracked across product release stages. While some differences in the definitions of Epic and Feature for Hybrid and Emergent work, the states are applied the same across work item types.

Epic States

| State | Definition | Transitions | Notes |
|-------------|---|---|-------|
| Not Started | Remains Not Started until work has begun at the Team level for any child item(s) | Product Management changes state to InProgress, Closed or On Hold | |
| In Progress | When the team is actively working on any child item(s) of an Epic it is "In Progress" | Product Management changes state to Closed or On Hold | |
| Closed | When all child work items are complete and there are no plans to add new items. | Product Management will not change state from Closed | |
| On Hold | If (for any reason) work is stopped, status is changed to On Hold while work is not being done. | Product Management changes state to Closed or In Progress | |

Epic State Transitions

```
stateDiagram-v2 [*] --> NotStarted NotStarted --> InProgress InProgress --> Closed InProgress --> OnHold OnHold --> InProgress OnHold --> Closed Closed --> [*]
```

Feature States

| State | Definition | Transitions | Notes |
|-------------|--|---|---|
| Proposed | Work has been determined, still may require refinements to overall features | Prod Mgmt changes from proposed to Approved, could move to InProgress | This is independent of Paid feature, T4 funded, or other types of Feature. Approved typically represents a customer funded project feature. |
| Approved | Feature was approved by customer (if paid) or T4 and ready for user stories. | Prod Mgmt can change to In Progress or On Hold | Indicates that user stories can be created for this feature. Why would a T4 Feature ever be in this state? |
| In Progress | At least one User Story have been completed (requirements) and Feature (or children) is actively being developed | ProdMgmt can change to OnHold, Closed, Cancelled | |
| On Hold | If (for any reason) work is stopped, status is changed to On Hold while work is not being done. | ProdMgmt can change to Approved, InProgress, Cancelled | e.g. this might be due to customer's blocking, not providing data, etc. |
| Beta | ??? | ??? | This is no longer needed for Features, Features would be left InProgress by ProdMgmt |
| UAT | ??? | ??? | This is no longer needed for Features, Features would be left InProgress by ProdMgmt |
| Closed | All requirements are met, all child work items meet "Definition of | | |

| State | Definition | Transitions | Notes |
|-----------|---|------------------------------------|---|
| | Done" and the feature is released to Production | | |
| Cancelled | Determined work was not needed, or could be cancelled by customer (if paid) | Should not change state after this | Code may have been written, but is not releasable |

Questions:

- Should Development Team be changing any of the Feature level states? Does this impact DevOPs and deployments? Should state changes be made by DevOps.

- When are Features prioritized and planned in the Roadmap and how is this indicated in a status?

Feature State Transitions

stateDiagram-v2 [*]--> Proposed Proposed-->Approved : Customer Funded note right of Approved Customer Paid features should be scoped to a single Feature end note Proposed-->InProgress Approved-->InProgress Approved-->OnHold InProgress-->OnHold InProgress-->Cancelled InProgress-->Closed OnHold-->Approved : Wouldn't Blocked be better? OnHold-->InProgress OnHold-->Cancelled Closed --> [*] Cancelled --> [*]

User Story

| State | Definition | Transitions | Notes |
|----------------------|--------------------|---|-------|
| To Do | Work has not begun | Prod Dev changes to BizReqs, TechSpecs, | |
| Ready for Refinement | | | |

| State | Definition | Transitions | Notes |
|--------------------------|------------|-------------|---|
| Business Requirements | | | |
| Technical Specifications | | | If this was done in Tasks, then there may be no need for this. |
| In Development | | | |
| Blocked | | | |
| In Code Review | | | |
| QA | | | |
| Alpha | | | |
| Beta | | | |
| UAT | | | |
| Ready for QA | | | Why have this and QA? Elena mentioned she works in QA and Priya works in Ready for QA. Why is there a status for a single person? |
| Pending QA Deployment | | | |
| Pending Merge | | | |
| Closed | | | |

If you have multiple items within the same priority tier, refer to the Severity or Priority on the work item. As usual, if you are still uncertain, you should ask the Product Manager, Scrum Master, or Technical Program Manager.

Do we need a state that indicates "Business Requirements Under Dev" or does "Business Development" cover this?

User Story State Transitions

```
stateDiagram-v2 [*]-->ProdMgmt state : ProdMgmt { ToDo-->ReadyForRefinement
ReadyForRefinement-->BizReqs BizReqs-->TechSpecs : Requirements and Story Pointing note
right of BizReqs During the Refinement Meeting end note } state : ProdDev { TechSpecs--
>InDevelopment InDevelopment-->InCodeReview InCodeReview-->PendingMerge
PendingMerge-->ReadyForQA ReadyForQA-->QA QA-->InDevelopment : Defect created QA--
>Closed } Closed-->[*]
```

Last update: May 1, 2022