# Best practices

## Development Best Practices

Use pair programming when neccesary Do TDD.

 Explain how a regular TDD flow goes.

100% resources must be on cloud. Provisioned and deployable. Use IAC.

Write as much as reusable modules.

## QA best practices

1. Keep the test cycles shorter than 3 days. Need to consider this during planning.
2. Write and run NFRs.
   1. Automate them. On a story level or on an application level.
3. Write at least the happy path test cases before development starts.
4. Test with production-like data.

## Devops Best Practices

Can you deploy at the click of a button. Can you roll back at the click of a button.

Can you provision and release infrastructure at the click of a button.? Do you use IAC and version control for your infrastructure.?

## BA Best practices

An epic should be requirements cleared when taking into a sprint.

## Logging & Monitoring

## Code review comment structure

Comment title - [cycle name]

Commend body:

Changes done:

[Explain the changes in a pointed format]

eg:-

dev-test cycle 2 changes done:

 Added field x to the database to store data y.  Changed button color red to gray.

## Test case driven development

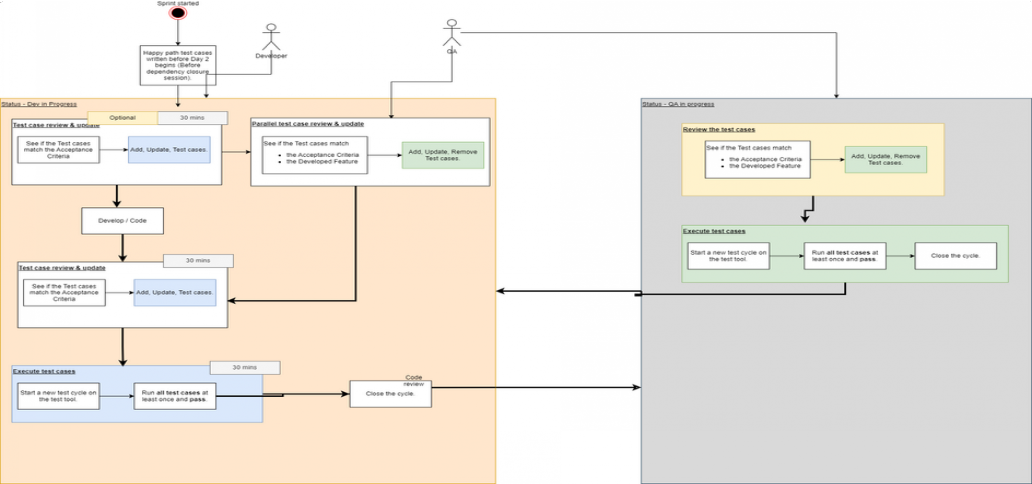
Set outlook alerts for Jira AIO assignee emails. Following is a flowchart explaining the process. Do dev testing in the local environment.

[Code review comment structure](https://inivos.atlassian.net/l/cp/VW43K1fD)

To be added to the process - add a comment in the above structure when doing a release. Or give a training for QAs to check github PR comments.



The cycle name should be [Dev/QA] - [sequence number 0 - n] - [some other information about the cycle]



## Triages

Do triages. Introduce a process around this concept. This can be used to make mid-sprint scope adjustment decisions.

## Default due dates per transition

These due dates will be set automatically by default. The assignee can then change the date manually to reflect the actual date of next transition.

→

**QA IN-PROGRESS**

**DONE**

If status changed to due by EOD

If status changed to

/

→ before 10:00

after 10:00

**QA IN-PROGRESS**

**DEV IN-PROGRESS**

**QA IN-PROGRESS**

**QA IN-PROGRESS**

(Under QA responsibility)

due by next day EOD.

→ If status changed to

**DEV IN-PROGRESS**

**DEV CODE-REVIEW**

**DEV CODE-REVIEW**

due by EOD

If status changed to

**DEV CODE-REVIEW**

due by next day EOD.

(Under lead responsibility) before 10:00

after 10:00

→ if SP = 1

**DEV TODO**

**DEV IN-PROGRESS**

 If status changed to

(Under dev responsibility)

before 10:00

**DEV IN-PROGRESS**

due by EOD

If status changed to

after 10:00

**DEV IN-PROGRESS**

due by next day EOD. if SP = 3

 If status changed to

**DEV IN-PROGRESS**

before 10:00

due by +1 day EOD. If status changed to

**DEV IN-PROGRESS**

after 10:00

due by +2 days EOD. if SP = 5

 If status changed to

**DEV IN-PROGRESS**

before 10:00

due by +1 day EOD. If status changed to

**DEV IN-PROGRESS**

after 10:00

due by +2 days EOD. if SP = 8

 If status changed to

**DEV IN-PROGRESS**

before 10:00

due by +2 day EOD. If status changed to

**DEV IN-PROGRESS**

after 10:00

due by +3 days EOD. if SP = 13

 If status changed to

**DEV IN-PROGRESS**

before 10:00

due by +4 day EOD. If status changed to

**DEV IN-PROGRESS**

after 10:00

due by +5 days EOD.

Any → (Under TPM responsibility)

**ON HOLD**

If status changed to  due by EOD

If status changed to

before 10:00

after 10:00

**DEV CODE-REVIEW**

**DEV CODE-REVIEW**

due by next day EOD.

Assumptions

We are currently not setting due dates for Dev todo & QA todo statuses.

Automated due date defaults are a rough estimation. The actual due date could be largely different from the automated setting. Team members can update the due date after automation sets due dates to reflect the actual date.