

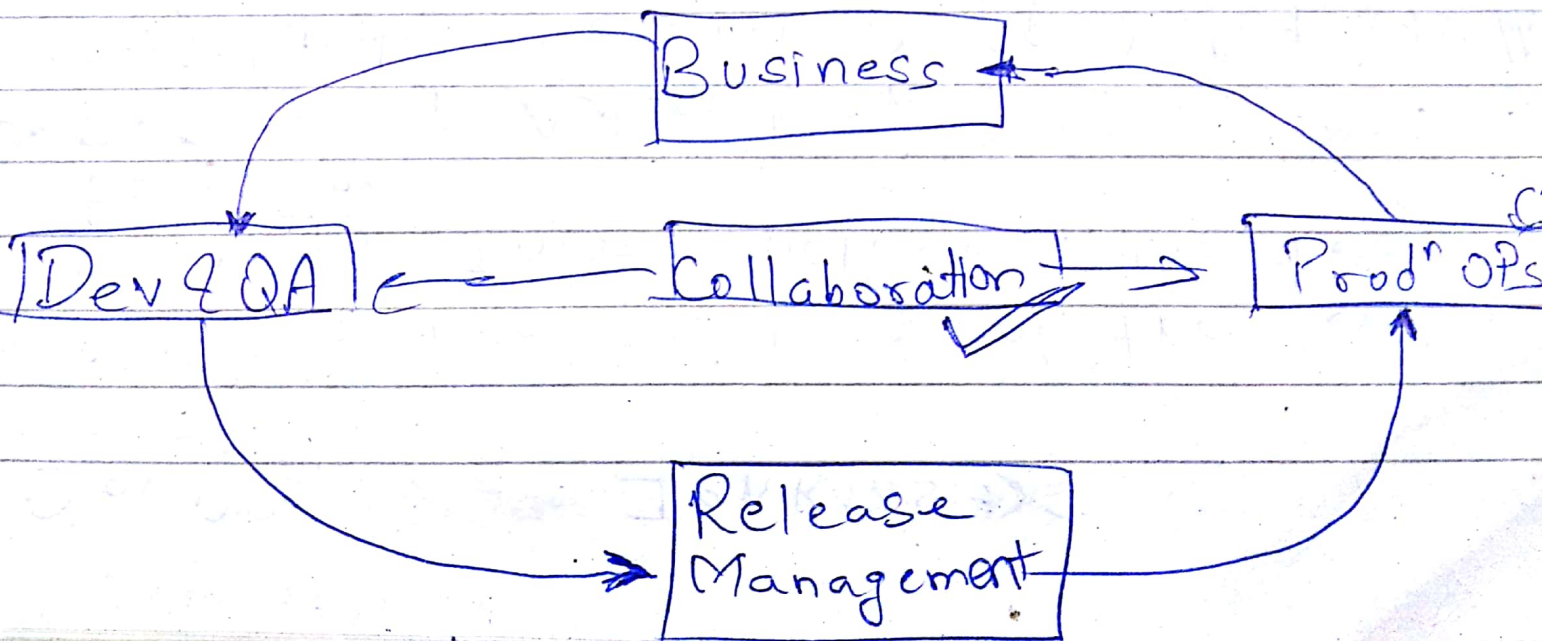
DevOps and the building blocks

⇒ WebApps & Mobile Apps.
↳ Developing & Deploying.

Synchronous mode. { Dev → Make changes in the code.
Ops → Deploy a quality sw
↳ Regular release of high qly sw

* Knowledge Transfer & SLA's are met.

↳ from Dev team to Ops team
↳ Deploy & then support it..



* DevOps ✓ Jenkins → ✗

↳ Collaborative working of the Development, testing, Infrastructure, ops teams together.

→ Driven by Automation.

→ Maturity measured → speed, freq, stability, & qly

* Infy DevOps

↳ Integrated Dev + Ops → 1 team now
↳ Unified process

Dev → Kanban

↳ Scrumban

Integration testing ⇒ Flag End of project
↳ Hell
↳ More time than devel

Continuous Delivery ⇒ Integration Pipeline. Subsy

SLA Agreement can be made faster.

Ops L1, L2 tickets \Rightarrow Automated \Rightarrow Repeated

CVS \Rightarrow Centralized version control
①
② Distributed.

* Only one person at a time can update

Lot of limitations in centralized version control
 \rightarrow Predictive monitoring.

* Modularizing of code

Tools \Rightarrow SONAR \rightarrow Test \Rightarrow Automation
 \hookrightarrow Open source

SQALE \Rightarrow S/W Qlty Assessment Lifecycle Expectation
(based on)

* Technical Debt \rightarrow Cost \uparrow over time
* redundant \Rightarrow Not affected

Code Coverage \rightarrow unit testing \rightarrow Automate testing

Maven & Ant \Rightarrow Automation tools
Test Scripts
Jenkins \Rightarrow Git

MD Pai
→ Bly 12

* KANBAN

→ CCD

- ↳ Visual board
- ↳ Visualization of workflow.

Final → Delivery

Deployment → from testing env to pre-prod
Env top

Del → Mgt perspective

Deployment → Administrative

* Refactoring

fungible codes

↳ New Release

↳ Fit to the right.

→ Frequent manual release.

→ Freq. KT from Dev

→ Post prod support.

Lab on cloud → slot

Dev team → Cont Int
Ops team → Incident mgt.
Test → Progress
Ops → Policies

• BIZ → Big room plan

CJ → Versioning
Static Code
Code coverage } Automation

001 .

* AGILE SCRUM

→ AD-COE

* Unit test cases. ⇒ Acceptance criteria
↳ Correctness of functionality.
↳ Alternate paths, validations.

* Scrum Master

↳ Remove impediments.

User Story ⇒ Story points ⇒ Estimation done by PO
in Sprint Zero.

Sprint review - Demo to PO
& End users.

↳ Discovery phase.

→ Burned down charts

Metrics

↳ Earned Biz value

Cum. Biz. Value

Story → Story Pts

Task → hrs.

Time bound, CIP, work together

AGILE SCRUM CEREMONIES

⇒ ① Sprint planning meeting
Team capacity

Defⁿ of Done → Activity from sprint Backlog is done.

② Daily Scrum ⇒ To speak about impediments

③ Sprint Review (Demonstration) meeting

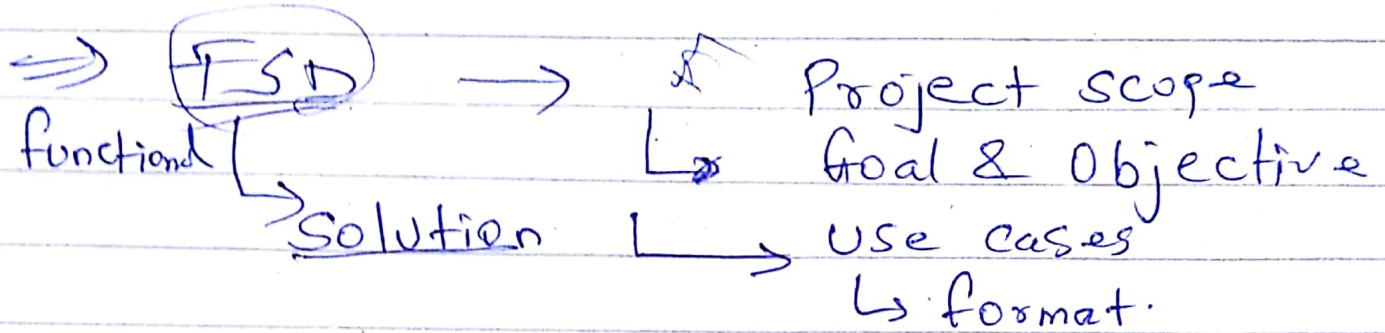
⇒ Demo of working software & assessing the feedback

④ Sprint Retrospective Meeting

⇒ To brainstorm & agree on what is working and what is not

* BRD

Components



Primary, Secondary actor
Exception flow

Use cases

* Format

FSD

Tools
→ Data Flow Diagram
→ Func Flow Diagram

Fields

BRD → High level

* Exception Handling

Overview of FSD

Not Solution

⇒ FSD derived from BRD
→ Address in which variables are stored
→ functions

* User story format

↳ Acceptance criteria.

* User story ⇒ short, simple descr.

Large user story ⇒ Epic
↳ Multiple smaller user stories

Acceptance criteria ⇒ Cond that s/w product must satisfy to be accepted by a user.