**Day 8: Test Closure and Production Release**

Is sufficient testing done?

* The answer - it depends - on delivery deadlines, risks, hign fixing costs, etc.
* When to say sufficient testing is done?
  + All of our tests have been executed, including the testing of new changes and regression.
  + All of it high risk areas are covered
  + No blocker or critical defects are open.
  + Product is stable, and low defect detection
* What affects sufficient testing?
  + Management decision, an approaching release deadline, cost of fi

Test Closure

* Test closure activities are those performed at the end of testing process.
* Even though testing is completed, these activities are important for future efficiency and should not be ignored.
* This commonly includes following activities (varies by organization):
  + Check for completion of tests – ensure all test work completed, i.e., all tests runs, all defects resolved/deferred in agreement
  + Hand-over – work products should be passed on to designated teams, example open defect log to product team, setup information to system teams
  + Lessons learned – document all learning, best practices, feedback etc
  + Archive documents – test plan, reports, logs etc

Product release stages

* Before release, product/development/system/QA team get together for “Go/No-Go” Meeting
* The meeting is usually led by Project Manager
* The objective is a collective decision to release or not
* If everyone is a “Go” then only product release is approved. Discussion moves on to deployment plan and schedule
* If even one party is a “No-Go” the release is not approved. The no-go party has to provide reason and a collective plan is made how to turn “Go”
* QA team will vote “Go” if the planned testing is completed, and there are no open critical/blocker defects
* The next step is to go over deployment plan
* The plan involves system admins (to deploy), Development team (to check logs), QA (to test new features/regression/sanity), Product team (to UAT)
* There should always be a “back-up” plan in case of any issues
* Releases should be planned such that there is no or minimal downtime
* If at all downtime is not avoided (for internal customers), a maintenance message should be put up before-hand and communication sent to clients
* If the released feature is not being rolled out to all customers, testers have to verify all versions (new + old)