SOFTWARE QUALITY – software product meets the gathered requirements
SOFTWARE QUALITY ASSURANCE – define and assess the adequacy of process
REPUTATION – organization rely on this. Impacts on clients or customer
LIMITING TECHNICAL DEBT – expensive to develop and maintain
SOFTWARE CERTIFICATION – software might require some form of certification
LEGALITY – legal obligations…..that use the software
ETHICAL CODES OF PRACTICE – system is not covered by software certification….engineers should do to maximize the quality of software
SOFTWARE PROTOTYPING – display the functionality of the product under development

(PROTOTYPING PARADIGMS)
THROWAWAY PROTOTYPING – fast method of prototyping….discarded after testing
EVOLUTIONARY PROTOTYPING – reused after testing and after obtaining feedback….used in late development stages

LOW FIDELITY PROTOTYPING – check and test the visual appearance and user flows of a software…..throwaway prototype can be created in low fidelity
PAPER BASED PROTOTYPING – used to create prototype based on hand drawings
WIREFRAMING – visual representation of a product page….used to create interactive prototype (BALSAMIQ WIREFRAME, AXURE, MOCKFLOW, ADOBE XD)
HIGH FIDELITY PROTOTYPING – created when developers have solid understanding of what they are going to build

SOFTWARE QUALITY ASSURANCE – structured approach to improve software quality
SOFTWARE TESTING AND INSPECTION – two commonly used methods to detect defects in software
SOFTWARE INSPECTION AND REVIEWS – concerned with the review of software artifacts

SOFTWARE INSPECTION OR REVIEWS – formalized peer review process
SOFTWARE ARTIFACTS – things that have been documented and stored

FAGAN INSPECTION METHODOLOGY – developed by Michael Fagan to identify and remove errors in the software products
MODERATOR – leads the inspection team and takes care of logistics
AUTHOR – the creator of the software artifact under inspection
READER – experienced peer can be a subject matter expert
TESTER – responsible for executing test cases for the software module

7 ACTIVITIES FOR FAGAN INSPECTION METHODOLOGY

PLANNING (Moderator**)** – identify inspectors and roles, verify the materials ready, distribute inspection mats, arrange meeting place and time
OVERVIEW(Author**)** – brief participants on information
PREPARATIONS(Inspectors**)** – prepare for the meeting, facilitate the preparations, read through deliverables
INSPECTION MEETING(Moderator/Inspector**)** – sets the time limit, perform their roles, record any defects, inspection outcome is agreed
PROCESS IMPROVEMENT(Inspector**)** – improve the development, record the cause analysis
REWORK(Author**)** – corrects the defects
FOLLOW-UP (Moderator/Author**)** – moderator verifies that the author has resolve the investigations

INSPECTION TYPE

REQUIREMENTS – review each page of requirements and raise question or concerns
DESIGN INSPECTION – review each page of design, compare it to requirements, and raise questions or concerns
CODE INSPECTION – review the code, compare it to requirements and/or design, and raise question or concern

STRUCTURED WALKTHROUGH – it is a peer review in which the author of a deliverable…its objective are to get feedback from the reviewers on the quality

STEP 1 – author circulates the deliverable
STEP 2 – author schedules a meeting with reviewers
STEP 3 – reviewers familiarize themselves with the deliverable
STEP 4 – review leader chairs the meeting
STEP 5 – author brings the review audience through deliverable
STEP 6 – scribe record error, decision, and any action items
STEP 7 – meeting outcome is agreed
STEP 8 – deliverable is circulated to reviewers
STEP 9 – author or project leader stores the comments and sign-offs

CODE REVIEWING TECHNIQUE

CODE REVIEWING/ INSPECTION – act of systematically convening with fellow programmers to check each other's code for fault to ensure meet quality standard
TOOL-DRIVEN CODE REVIEW – uses automated code analysis techniques
DEVELOPER-DRIVEN CODE REVIEW – the developer themselves reviews the code to pick out the problems.

MODERN CODE REVIEW

MODERN CODE REVIEW – lightweight variant of the code inspection does not rely face-to-face meeting
TOOL-DRIVEN CODE REVIEW – Google Gerrit and Microsoft code flow. a repository is a central file storage location
PULL-BASED DEVELOPMENT – enforce modern code review does not use code reviewing tools "pull request"