SOFTWARE QUALITY ASSURANCE (SQA) – AN OVERVIEW

PROGRAMMING APPLICATIONS AND FRAMEWORKS (IT3030)

LEARNING OUTCOMES

- After completing this lecture, you will be able to,
  - Define what Software Quality Assurance mean.
  - Differentiate between SQA and testing.
  - Discuss the importance of SQA in today's software industry.
  - Apply SQA practices in your development work.

CONTENTS

- What is Software Quality?
- Why is Software Quality Assurance (SQA) Important?
- The Role of SQA in the Software Development Life Cycle (SDLC)
- SQA Activities and Processes
  - The SQA Process
  - Building Quality into the Software Development Process
- SQA, Agile and DevOps
- Self-study activity
- Summary

WHAT IS SOFTWARE QUALITY?

- According to the IEEE Computer Society, software quality refers to,
  - how well a software product **conforms** to the given **requirements** and **meets** the **needs** of its **users**. It involves both the software **product itself** as well as the **processes** used to develop it (source).

WHAT IS SOFTWARE QUALITY?

- The American Society for Quality defines Software Quality Assurance (SQA) as,
    - A **systematic approach** to evaluating the quality of and adherence to software product standards, processes and procedures. SQA includes **ensuring standards** and **procedures** are **established** and **followed** throughout the software acquisition life cycle (source).

WHY IS SOFTWARE QUALITY ASSURANCE (SQA) IMPORTANT?

- SQA is a crucial part of the software development process, **ensuring** a **final product** that meets **expectations**, that involves **people**, **processes**, and **technology**.
- SQA leads to a **positive user experience** by minimising failures and ensuring conformity with the user requirements. This translates to satisfied users who are more likely to recommend the product.

WHY IS SOFTWARE QUALITY ASSURANCE (SQA) IMPORTANT?

- SQA helps businesses **save money, time and reputation** by preventing or catching defects early. Fixing bugs later in the SDLC is significantly more expensive in all three aspects.
- SQA helps businesses **comply with industry standards and regulations**, reducing legal risks.

WHY IS SOFTWARE QUALITY ASSURANCE (SQA) IMPORTANT?

- SQA **emphasizes documentation**, ensuring that the requirements, processes, design decisions, technical information, test results and reports are well documented, which aids in understanding the base code and **makes maintenance easier**.

WHY IS SOFTWARE QUALITY ASSURANCE (SQA) IMPORTANT?

- SQA is **NOT** just testing!
  - SQA is a process that focuses on preventing defects through **proactive** measures throughout the SDLC and ensures overall quality.
  - Software Testing is a specific activity within SQA. It's a **reactive** approach that helps find defects, but SQA goes beyond testing to prevent them from occurring in the first place.
  - Read more on proactive vs reactive approaches to quality here.

Source: https://unsplash.com/photos/black-red-and-white-textile-hbb6GkG6p9M

WHY IS SOFTWARE QUALITY ASSURANCE (SQA) IMPORTANT?

- Real-world examples of software failures
  - Therac-25 (6 deaths due to radiation overdose)
  - Boeing 737 Max Crashes (346 deaths, US$20-80 Billion in costs)
  - Y2K Bug (US$300 billion)/ Year 2038 problem (yet to occur)
  - Loss of Mars Climate Orbiter (US$93 million)
  - Ariane 5 (US$8 million)
  - …how did your ITP project go? ☺

Source:
https://en.wikipedia.org/wiki/Year_2000_problem#/media/File:Bug_de_l'an_2000.jpg

THE SQA PROCESS

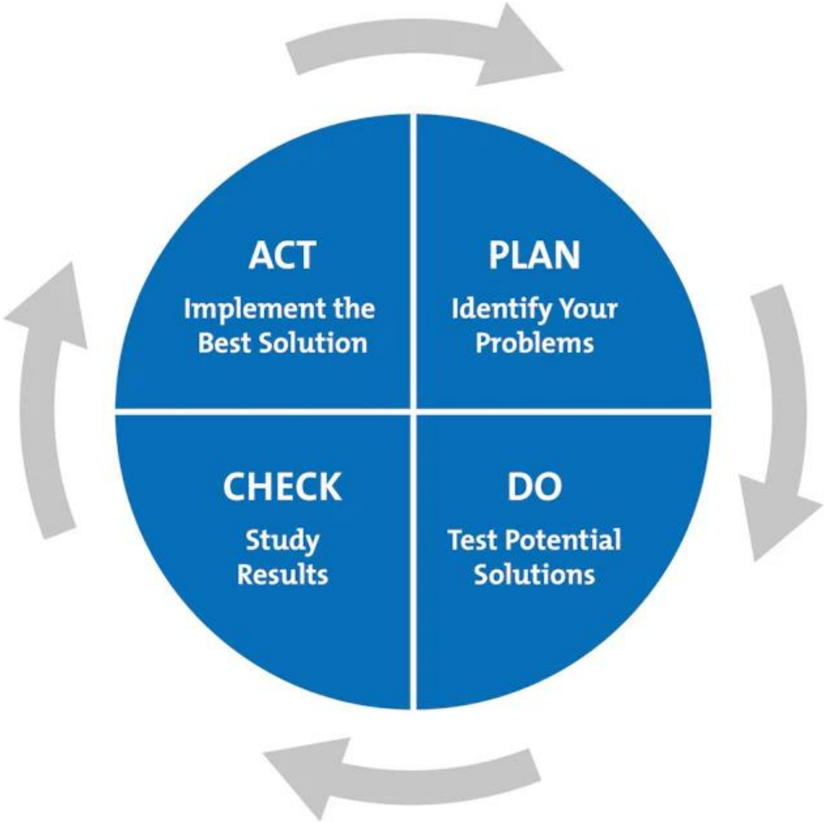
- Identify Requirements
- Establish quality standards based on the requirements
  - Clearly define the quality standards that the software product must meet. This includes defining requirements, acceptance criteria, and performance metrics.
- Plan SQA activities
  - Process for problem reporting and corrective action
  - Software artefact management
  - Process reviews/ audits
  - Testing methodology

THE SQA PROCESS

- Execute the SQA plan
  - Conduct reviews of software artefacts
  - Perform testing (repeatedly) – use automation to increase efficiency.
  - Conduct process reviews
- Monitor the quality of the product
  - This should be done throughout the SDLC.
    - Defects must be tracked
    - Analyze metrics such as code coverage and defect density
    - Conduct root cause analysis

THE SQA PROCESS

- Continuously improve the SQA process
  - Use the findings from the process reviews and monitoring data to identify areas for improvement and implement changes to the SQA process.
  - Follow the Plan-Do-Check-Act (PDCA) Cycle.



Source: https://www.mindtools.com/as2l5i1/pdca-plan-do-check-act

BUILDING QUALITY INTO THE SOFTWARE DEVELOPMENT PROCESS

- Define Clear Requirements
  - Start by clearly defining and documenting the requirements for the software. This ensures that everyone involved understands what needs to be built and what constitutes success. Effective requirement management is critical for delivering a successful software product.
- Adopt Agile Methodologies
  - Agile methodologies like Scrum or Kanban promote iterative development, frequent collaboration, and continuous improvement. This allows for early detection and resolution of issues, leading to higher-quality software.

BUILDING QUALITY INTO THE SOFTWARE DEVELOPMENT PROCESS

- Design Reviews
  - Design reviews involve a methodical examination of the software design. The goal is to identify and address any defects, inconsistencies, or areas for improvement in the design before coding begins. This proactive approach helps to prevent errors from being coded into the software and reduces the time and effort required for fixing problems later in the development process.
- Implement Code Reviews
  - Regular code reviews help catch issues early in the development process. They also facilitate knowledge sharing among team members and ensure adherence to coding standards and best practices.

BUILDING QUALITY INTO THE SOFTWARE DEVELOPMENT PROCESS

- Automate Testing
  - Automated testing, including unit tests, integration tests, and end-to-end tests, helps identify defects quickly and consistently. Continuous integration and continuous deployment (CI/CD) pipelines can automate the testing process, enabling faster feedback loops.
- Use Version Control
  - Version control systems like Git enable teams to track changes to the codebase, collaborate effectively, and revert to previous versions if needed. This promotes transparency and reduces the risk of introducing errors.

BUILDING QUALITY INTO THE SOFTWARE DEVELOPMENT PROCESS

- Prioritize Security
  - Incorporate security practices throughout the development process, including threat modelling, code scanning, penetration testing, and security reviews. Security should be considered at every stage, from design to deployment.
- Focus on User Experience (UX)
  - Design the software with the end user in mind. Conduct user research, gather feedback, and iterate based on user needs and preferences. A well-designed user interface enhances usability and overall satisfaction.

BUILDING QUALITY INTO THE SOFTWARE DEVELOPMENT PROCESS

- Document everything
  - In addition to the standard documentation (SRS, test plans etc.), document the thought process behind technical decisions. This should be a continuous process. This enables proactive identification of any issues during reviews as well as making it easy for anybody new to get up to speed quickly.
- Continuous Monitoring and Feedback
  - Implement monitoring and logging mechanisms to track system performance, errors, and user behaviour in real time. This enables proactive identification of issues and opportunities for optimization.

BUILDING QUALITY INTO THE SOFTWARE DEVELOPMENT PROCESS

- Invest in Training and Education
  - Provide ongoing training and professional development opportunities for team members to stay updated on emerging technologies, best practices, and industry trends. Knowledgeable and skilled teams are better equipped to deliver high-quality software.
- Promote a Culture of Quality
  - Foster a culture where quality is everyone's responsibility. Encourage open communication, collaboration, and accountability among team members. Recognize and reward efforts that contribute to maintaining and improving software quality.

SQA, AGILE AND DEVOPS

- Traditional approach (Waterfall SDLC)
  - It is a sequential process where each phase of the software development lifecycle is completed before moving on to the next phase.
  - SQA is performed at the end of each phase to ensure that the requirements have been met before moving to the next phase.
  - This approach involves requirement analysis, design, coding, testing, and maintenance to ensure that the software product is developed with minimal errors and defects and meets the desired quality standards.

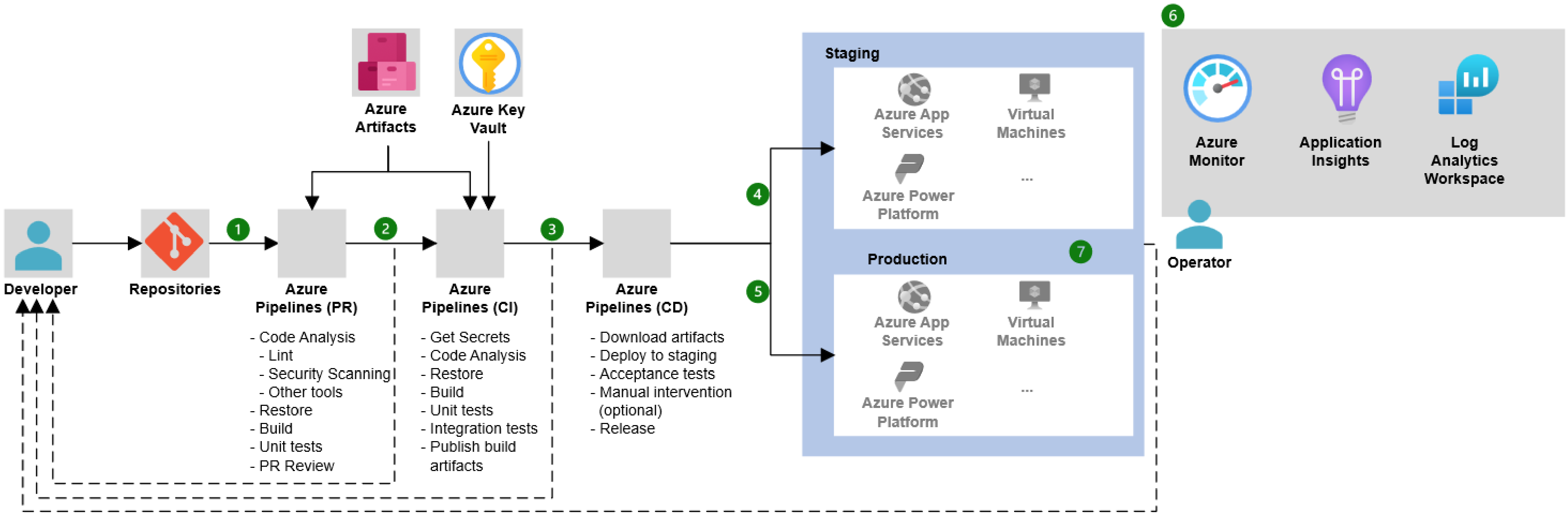
SQA, AGILE AND DEVOPS

- Agile approach
  - The Agile approach to SQA is an iterative, incremental, and flexible approach that focuses on delivering software products in small increments.
  - This approach emphasizes **collaboration** between the development team and the stakeholders for a seamless and quick development process.
  - Agile SQA is quite popular and focuses on self-organizing teams, continuous integration and testing, continuous delivery, and continuous feedback to ensure a high-quality software product.

SQA, AGILE AND DEVOPS

- DevOps approach
  - This approach promotes **collaboration** between development and IT operations to ensure that the software product meets the requirements of the customers.
  - DevOps is focused heavily on automation, such as continuous integration, continuous testing, and continuous deployment to deliver a high-quality product.
  - This approach is great for projects that require frequent updates.

EXAMPLE CI/ CD WORKFLOW ON AZURE DEVOPS



Source: https://learn.microsoft.com/en-us/azure/architecture/example-scenario/apps/devops-dotnet-baseline

SELF-STUDY ACTIVITY

- Refer to the following material and identify how SQA could have prevented them.
  - CASE STUDY - THERAC-25
  - HOW THE BOEING 737 MAX DISASTER LOOKS TO A SOFTWARE DEVELOPER

SUMMARY

- What is Software Quality?
- Why is Software Quality Assurance (SQA) Important?
- The Role of SQA in the Software Development Life Cycle (SDLC)
- SQA Activities and Processes
  - The SQA Process
  - Building Quality into the Software Development Process
- SQA, Agile and DevOps

REFERENCES

1. What is Software Quality?
2. What Is Software Quality Assurance? A Step-By-Step Guide And Importance
3. What Is Software Quality Assurance, and Why Is It Important?

THANK YOU

VISHAN.J@SLIIT.LK

NELUM.A@SLIIT.LK

SASINI.H@SLIIT.LK