CS3219

AY24/25 Sem 1

github.com/jasonqiu212

01. Introduction

Software Types

- **Edge Computing** - Computation done at leaf nodes
- **Cloud Computing** - Host software on ext. data center
  - **Cloud-enabled** - Legacy applications modified to run on the cloud (vs. cloud-native)

Software Development Process

- **Waterfall** - Sequential approach good for stable req.
- **Agile** - Iterative development with feedback loops and quick responses to changes
  - **Scrum** - Work done in sprints, where a subset of the product backlog is cleared

Software Delivery

- **Deployment** - Make software available to use after dev.
  - Bare metal: Customized build for target platforms
  - Virtual machine: Use VM to run guest OS to run app.
  - Containers: Include only necessary OS processes and dependencies (Lighter than VM)
  - Serverless: Cloud-native servers that don't need developers to manage (Let provider manage resources)

DevOps - Practices combining software dev. and ops.

- Purpose: Reduce time between committing change to the change reaching production while ensuring quality
- **Cont. Integration** - Auto build, unit test, deploy to staging, and acceptance test, to show problems early
- **Continuous Delivery** - Same as above, except with manual deployment to production. Ensures that every good build is potentially ready for production release.
- **Continuous Deployment** - Same as above, but with auto deployment to production

02. Requirements

- **Requirement** - Capability needed by a user or must be met by a system

Types of Requirements

- **Business Req.** - Why the organization is implementing the system, e.g., reduce staff costs by 25%
- **User Req.** - Goals the user must be able to perform with the prooduct, e.g., check for flight on website
- **Functional Req. (FR)** - Specifies what a system does, e.g., website can export boarding pass
- **Business Rules** - Policies that define or constrain requirements, e.g., staff gets 40% discount
- **Quality Attributes** - How well the system performs, e.g., Mean time bet. failure $\geq$ 100 hours. A type of non-functional req.
- **System Req.** - Hardware or software issues, e.g., invoice system must share data with purchase order system
- **External Interfaces** - Connections between systems and outside world, e.g., must import files in CSV format
- **Constraints** - Limitations on implementation choices, e.g., must be backward compatible. Type of NFR.
- Flow: Business Req. $\rightarrow$ **Vision and Scope Document** $\rightarrow$ User Req. $\rightarrow$ **User Req. Doc.** $\rightarrow$ FRs $\rightarrow$ SRS

Requirements Development Phases

- **Elicitation** - Discover requirements (e.g., Interview)
- **Analysis** - Analyze, decompose, derive, understand
- **Specification** - Written or illustrated requirements
- **Validation** - Confirm correct set of requirements
- No linear path

Requirements Development Outcomes

- **Software Req. Specification (SRS)** - Complete desc. of behavaior of software. Contains FRs, System Req., Quality Attributes, Ext. Interfaces, and Constraints.
- **Rights, Responsibilities, and Agreements** - All stakeholders confident of development within balanced schedule, cost, functionality and quality
- **Change Control** - Process to ensure changes to a product are introduced in a controlled and coordinated way

Quality Attributes

External

Internal