Course ID: CSE - 471
Course Title: System Analysis and Design

Course Coordinator:

Dr. Muhammad Iqbal Hossain
Associate professor
Department of Computer Science & Engineering
BRAC University.

Guideline for students

- Every student need to attend the online lectures in every week.
- While proceeding with the video lectures there might be pop-up quizzes, segment quiz etc. You need to participate in those quizzes to proceed to next lecture. **Please note that these quizzes will be used for evaluation**. So listen the lectures carefully and take the quiz.
- Faculties will be available ONLINE (google meets) in the class time (according to the USIS) where students can join and discuss different topics, difficulties etc. You will be provided the link. Make sure you have finished the online lecture before joining the class meeting.
- Assignment needs to be submitted by the provided deadline.

Resources

- Text book: **System analysis and design** 5th edition, Dennis, Wixom, Roth.
- Presentation slides
- Other resources will be shared between the lecture.

Lecture 1

Segment 1: What is System analysis and design

Segment 2: Software lifecycle

Segment 3: DevOps

Segment 4: DevOps (cont.)

Segment 5: Project Team Roles and Skills

Segment 1

What is System analysis and design

Information System

- Information systems are software applications which manage large amounts of data, share information etc.
- Most of the software out there is information systems software,
  - written in languages such as Java, C++, .NET and the like.

Components of IS

- Hardware- these are the devices like the monitor, processor, printer and keyboard, all of which work together to accept, process, show data and information.
- Software- are the programs that allow the hardware to process the data.
- Databases- are the gathering of associated files or tables containing related data.
- Networks- are a connecting system that allows diverse computers to distribute resources.
- Procedures- are the commands for combining the components above to process information and produce the preferred output.

Software Horror Stories

- Bank of America spent $23,000,000 on a 5-year project to develop a new accounting system. Spent over $60,000,000 trying to make new system work, finally abandoned it. Loss of business estimated in excess of $1,000,000,000

- The B1 Bomber required an additional $1,000,000,000 to improve its air defense software, but the software still isn't working to specification

- Ariane 5, Flight 501 The loss of a $500,000,000 spacecraft was ultimately attributed to errors in requirements, specifications and inadequate software reuse practices.

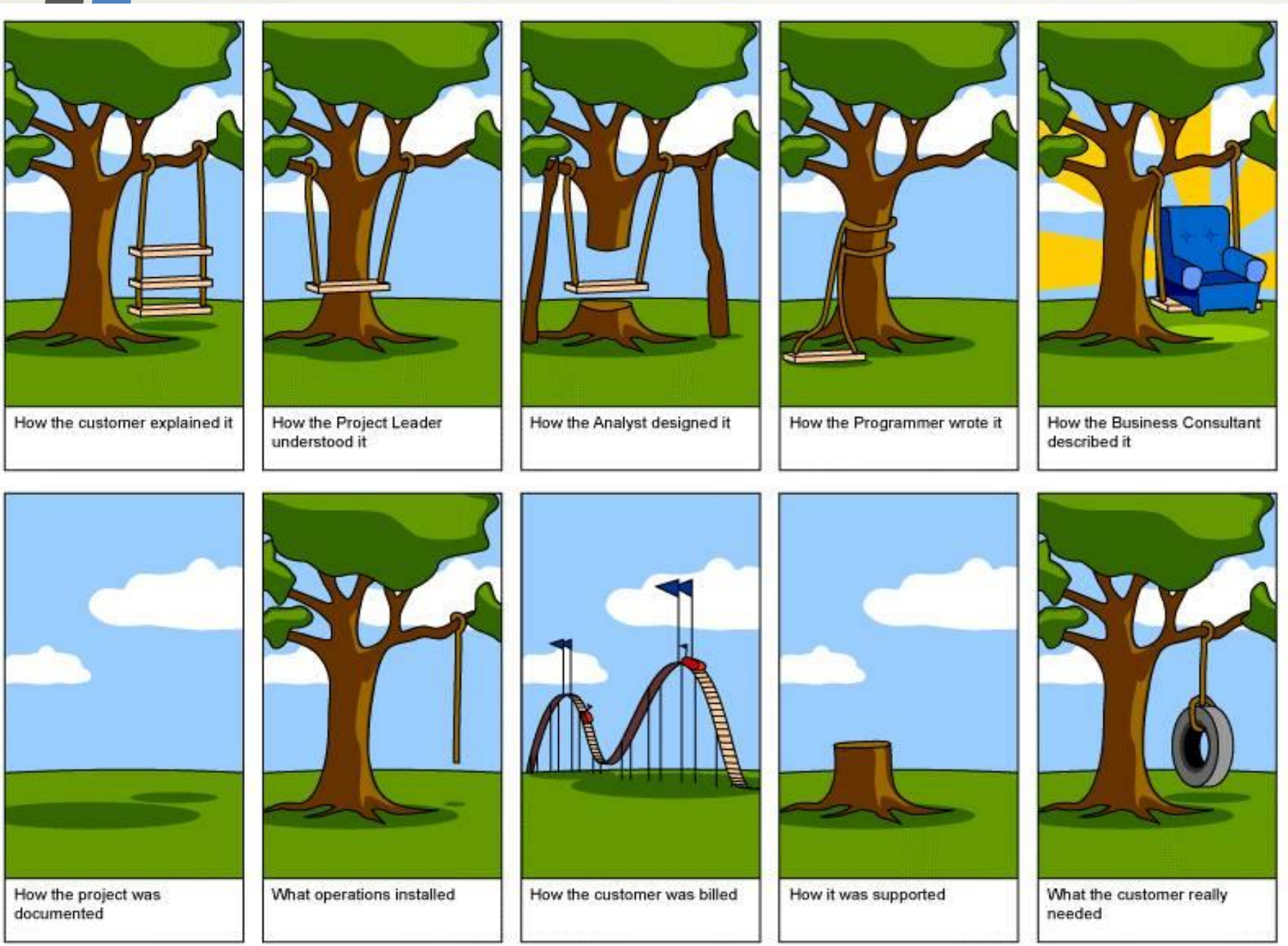
The Bad News

- 30% of large IT projects are **cancelled** before completion
- 50% of IT projects are **over-budget** by more than 200%
- The majority of completed projects deliver *60% or less of prescribed functionality*
- Many delivered information systems are **under-used** because they don't meet user needs and/or expectations
- Legacy systems are a serious and growing **bottleneck** to organizational evolution

Why is this Course Important?

- Most errors (54%) are detected after coding and testing.
- Almost half of all errors in software (45%) are in requirements and design.
- Most errors made during requirements analysis are non-clerical (77%)
- Requirements errors can cost up to 100 times more to fix than implementation errors
  - if they are not caught early on.

Need to do requirements and design right!


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Key Ideas

- Many failed systems were abandoned because analysts tried to *build wonderful systems without understanding the organization*.
- The primary goal is to create **value** for the organization.

Key Ideas

- **Systems analyst** is a key person
  - analyzing the business
  - identifying opportunities for improvement
  - designing information systems to implement these ideas.

What is System Analysis?

- The collection of notations, methodologies and tools used to gather details and analyze a problem situation prior to information system design and implementation

- ***Systems analysis*** (or, ***requirements analysis***) must ensure that the proposed information system meets **user needs**, can be delivered **on time**, and can be updated **inexpensively**.

- Problems in "getting the systems analysis right", such as ill-defined situations, ambiguities, inconsistencies, mixing requirements with design

Need for Systems Analysis?

- ***Remember, finding and fixing a fault after software delivery is 100x more expensive than finding and fixing it during systems analysis or early design phases***

End of Segment 1

Segment 2
Software lifecycle

Major Attributes of the Lifecycle

The project

- Moves systematically through **phases** where each phase has a standard set of outputs
- Produces project **deliverables**
- Uses deliverables in implementation
- Results in actual information system
- Uses *gradual refinement*

Project Phases

- **Planning**
  - Why build the system?
- **Analysis**
  - Who, what, when, where will the system be?
- **Design**
  - How will the system work?
- **Implementation**
  - System delivery

Planning

- Identifying business value
- Analyze feasibility
- Develop work plan
- Staff the project
- Control and direct project

Analysis

- Analysis strategy
  - Analysis of *current system*
  - Ways to design **new system**
- Requirements gathering
  - Interviews, questionnaires etc.
- Process modeling
- Data modeling

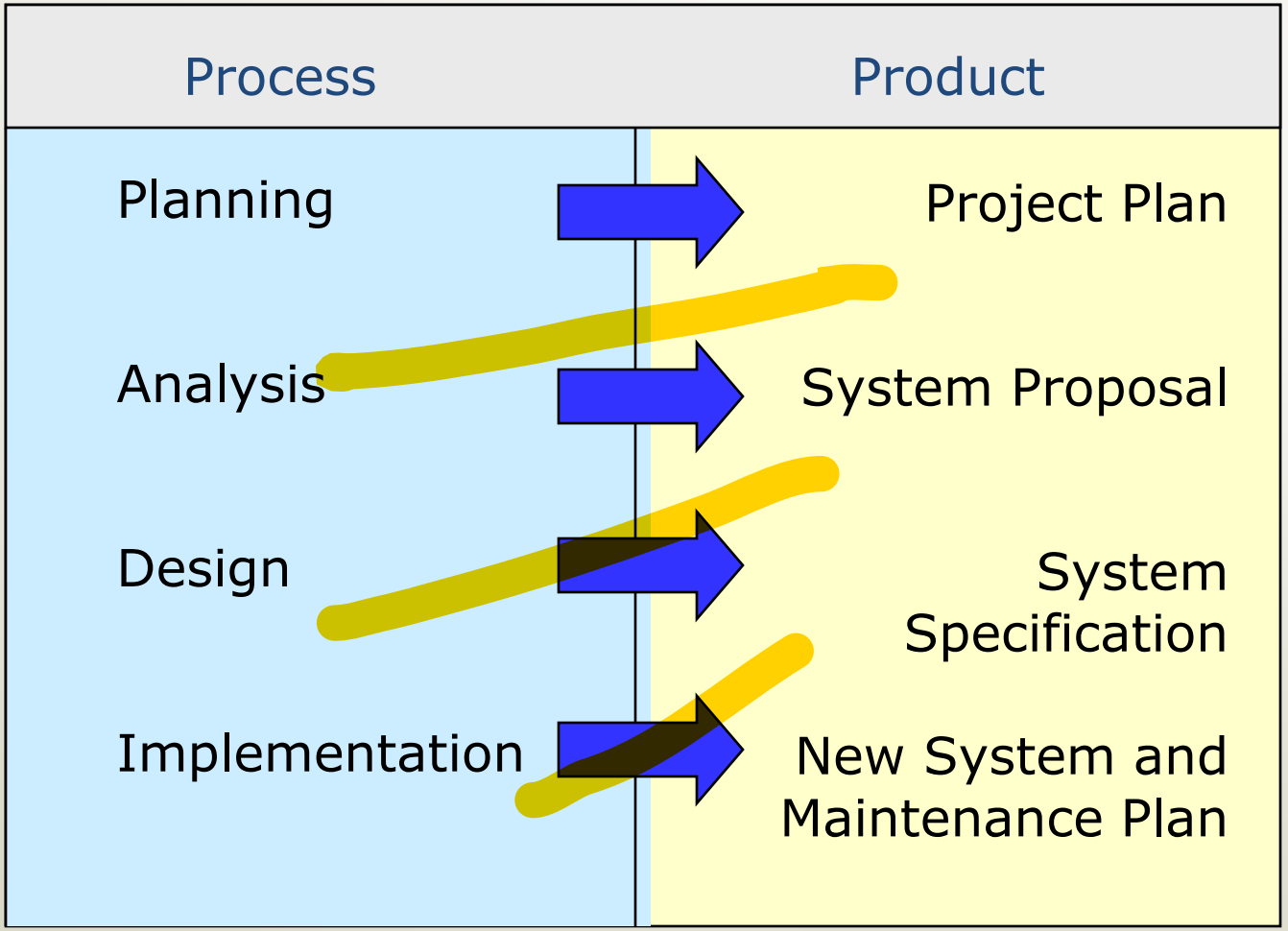
Design

- Architectural design
  - **Hardware**
  - **Software**
  - **Network** infrastructure
- Interface design
- Database and file design
- Program design

Implementation

- **Construction**
  - Writing programs
  - Testing
- **Installation**
  - Replace old with new system
  - Training users
- Support Plan

Processes and Deliverables



What Is a Methodology?

- **A formalized approach or series of steps to implement SDLC**

- Methodology categories:
  - Process-centered
  - Data-centered
  - Object-oriented

Need for methodology

- *Writing code without a well-thought-out system request may work for small programs, but rarely works for large ones.*

- Need to have a good **design** from the requirements before moving on to implementation.

Systems development methodologies

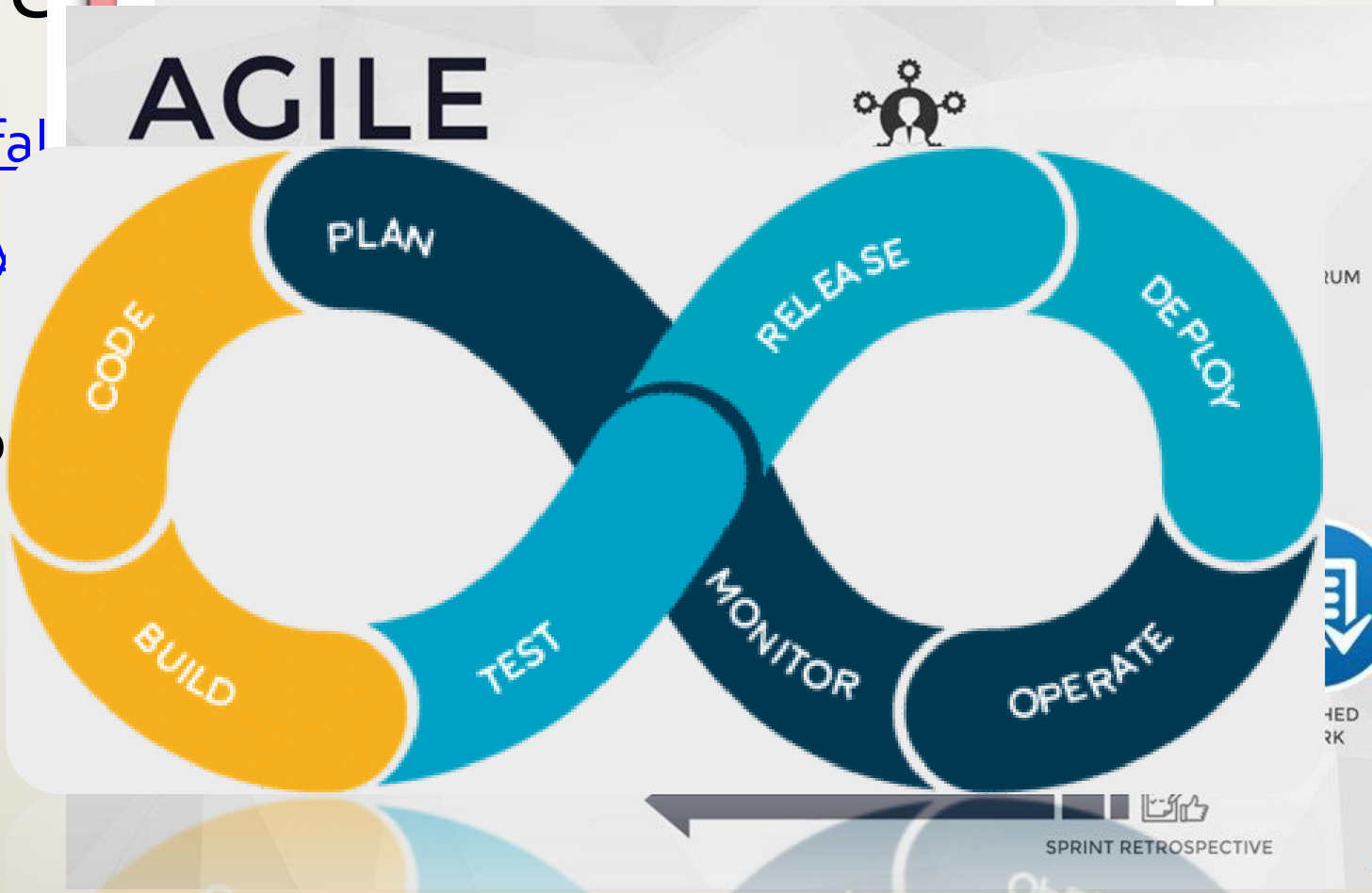
- **Structured Design**
  - Waterfall Development
  - Parallel Development
- **Rapid Application Development (RAD)**
  - Phased Development
  - Prototyping
- **Agile Development**
  - Extreme Programming
- **Devops**

End of Segment 2

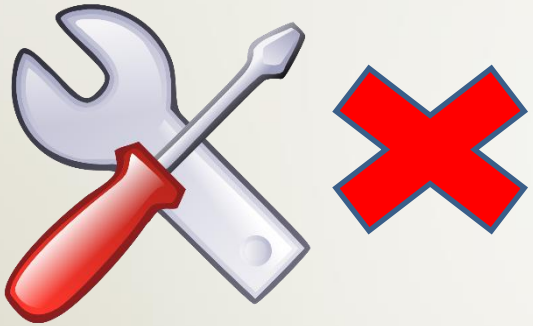
Segment 3
DevOps

Curre

- Waterfa
- prototy
- Agile
- DevOp



What is DevOps?



Why DevOps?

- Before DevOps, operation and development teams were working in an isolated environment.
- Testing and Deployment activities mostly were performed in an isolated manner after design-build step, and they took more time than actual project completion time.
- Team members usually spend a large amount of time in deploying, testing, designing, and building the projects

Why DevOps?

- Human production errors were deployed during manual code conduction.
- Operations and coding teams generally had different timelines and did not have proper synchronization that results in further delay.
- To avoid the hassles mentioned above and non-collaborative performance measures, there was an urgent need for robust IT technology like DevOps to satisfy business owners and stakeholders.

Current relation among Development and Operation team

- The development team worked on the software, developing it and making sure that the code worked perfectly.
- After hours of hardwork and a lot of trial and error, the team releases a code which has to be executed by the Operations team which is responsible for the release and operation of the code.
- The operations team will be checking the application and its performance and reporting back any bugs, if present.

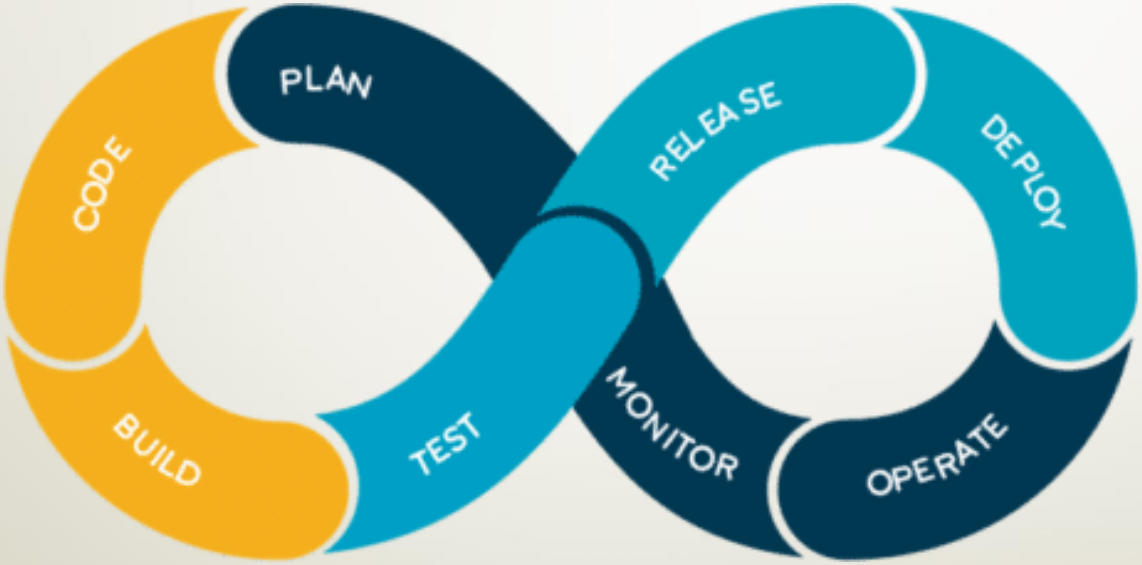
What DevOps can do?

- To bridge this gap, Development('Dev') team and Operations ('Ops') team collaborated giving rise to DevOps.
- The Development team encapsulated their code in a **container** which is a lightweight software environment.
- When the developers were done with their work, they would simply pass on this container along with the code to the operations team. The Ops will run this container, along with the code, and it worked as expected!

End of Segment 3

Segment 4
DevOps (cont.)

Steps of DevOps



Steps of DevOps

- **Code:** The first step in the DevOps life cycle is coding, where developers build the code on any platform
- **Build:** Developers build the version of their program in any extension depending upon the language they are using
- **Test:** For DevOps to be successful, the testing process must be automated using any automation tool like Selenium
- **Release:** A process for managing, planning, scheduling, and controlling the build in different environments after testing and before deployment

Steps of DevOps

- **Deploy:** This phase gets all artifacts/code files of the application ready and deploys/executes them on the server
- **Operate:** The application is run after its deployment, where clients use it in real-world scenarios.
- **Monitor:** This phase helps in providing crucial information that basically helps ensure service uptime and optimal performance
- **Plan:** The planning stage gathers information from the monitoring stage and, as per feedback, implements the changes for better performance

Everything should be Continuous

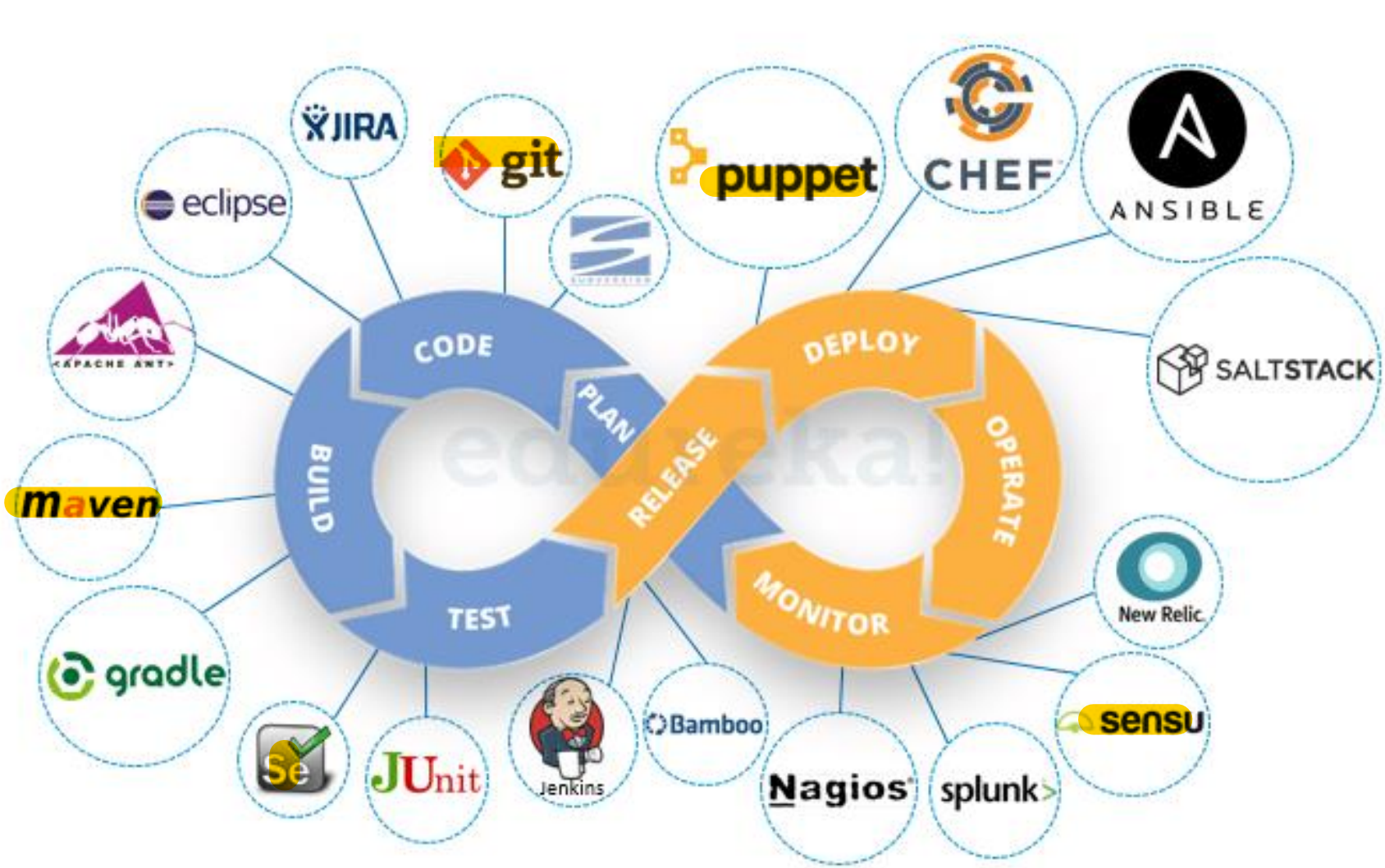
- **Continuous Development**
- **Continuous Testing**
- **Continuous Integration**
- **Continuous Deployment**
- **Continuous Monitoring**

The advantage of DevOps

- Reduced chance of product failure
- Improved flexibility and support.
- Faster time to market.
- Better team efficiency.
- Clear product vision within the team.

DevOps Challenges

- Difficulties with Integration
- Automated Testing
- Relatively High Costs
- Toolset Choice
- Lack of Talent


JIRA
git
eclipse
puppet
CHEF
ANSIBLE
CODE
PLAN
BUILD
TEST
RELEASE
DEPLOY
OPERATE
MONITOR
SALTSTACK
maven
gradle
New Relic
sensu
Bamboo
JUnit
Jenkins
Nagios
splunk

Segment 5
Project Team Roles and Skills

Information Systems Roles

- Business analyst
- System analyst
- Infrastructure analyst
- Change management analyst
- Project manager

Project Team Roles

Role	Responsibilities
Business analyst	Analyzing the key business aspects of the system Identifying how the system will provide business value Designing the new business processes and policies
Systems analyst	Identifying how technology can improve business processes Designing the new business processes Designing the information system Ensuring that the system conforms to information systems standards
Infrastructure analyst	Ensuring the system conforms to infrastructure standards Identifying infrastructure changes needed to support the system
Change management analyst	Developing and executing a change management plan Developing and executing a user training plan
Project manager	Managing the team of analysts, programmers, technical writers, and other specialists Developing and monitoring the project plan Assigning resources Serving as the primary point of contact for the project

Summary -- Part 1

- *The Systems Development Life Cycle* consists of four stages: Planning, Analysis, Design, and Implementation
- The *major development methodologies*:
  - Structured design
  - the waterfall method
  - Parallel development
  - RAD development
  - Prototyping (regular and throwaway)
  - Agile development
  - XP streamline SDLC

Summary -- Part 2

- There are five *major team roles*: business analyst, systems analyst, infrastructure analyst, change management analyst and project manager.