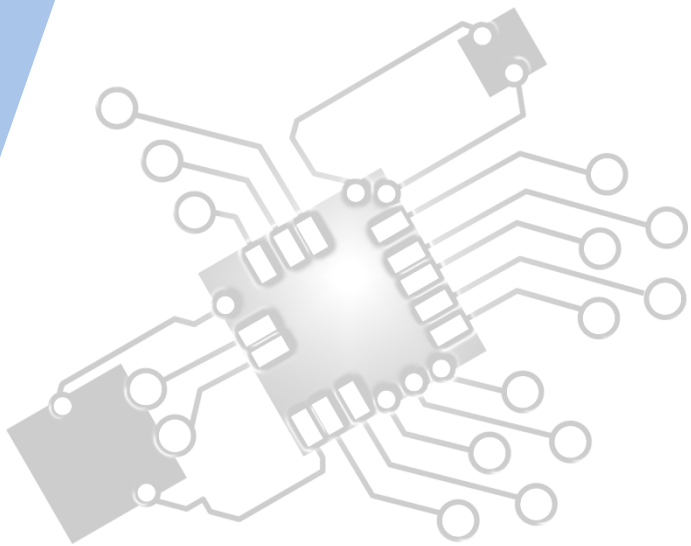
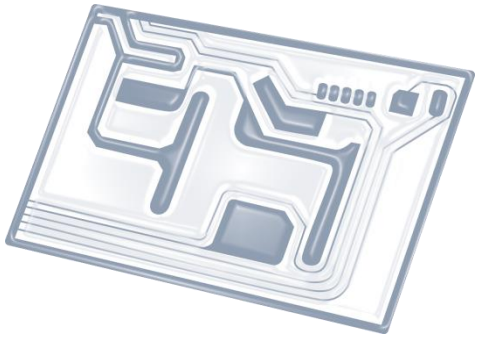
Software deployment

IB Computer Science



Content developed by
Dartford Grammar School
Computer Science Department

HL Topics 1-7, D1-4



1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP

HL & SL 1.1 Overview

Planning and system installation

1.1.1 Identify the context for which a new system is planned.

1.1.2 Describe the need for change management

1.1.3 Outline compatibility issues resulting from situations including legacy systems or business mergers.

1.1.4 Compare the implementation of systems using a client's hardware with hosting systems remotely

1.1.5 Evaluate alternative installation processes

1.1.6 Discuss problems that may arise as a part of data migration

1.1.7 Suggest various types of testing

User focus

1.1.8 Describe the importance of user documentation

1.1.9 Evaluate different methods of providing user documentation

1.1.10 Evaluate different methods of delivering user training

System backup

1.1.11 Identify a range of causes of data loss

1.1.12 Outline the consequences of data loss in a specified situation

1.1.13 Describe a range of methods that can be used to prevent data loss

Software deployment

1.1.14 Describe strategies for managing releases and updates

1: System design

2: Computer Organisation

3: Networks

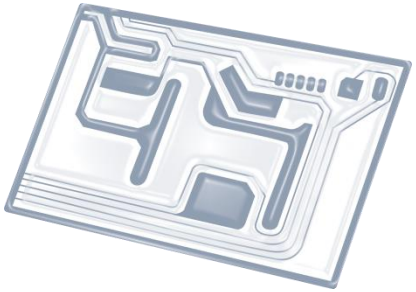
4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP



Topic 1.1.14

Describe strategies for **managing releases** and **updates**

Release

- A product release is the process of launching a **new product** for a specific market or user base.
- In software development, a product release is sometimes done with a beta version so that core developers/users can assist with debugging and feedback prior to the release of the actual software.

Update

- An update is a software file that contains **fixes** for problems found by other users or the software developer.
- Installing an update fixes the code and prevents the problems from happening on your computer.
- Because updates fix problems with a program, they are almost always free and available through the program or the companies website.

Patches

- A patch is a software update comprised code inserted (or patched) into the code of an executable program.
- Typically, a patch is installed into an existing software program.
- Patches are often **temporary fixes** between full releases of a software package.
- Patches may do any of the following:
  - Fix a software bug
  - Install new drivers
  - Address new security vulnerabilities
  - Address software stability issues
  - Upgrade the software

How to get updates

- **Manually** (download and install them yourself)
  - **Pro**: Full control
  - **Con**: Time consuming

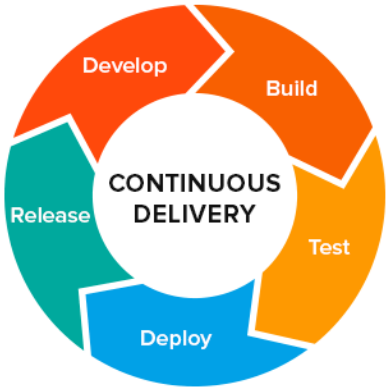
- **Automatically** (software automatically contacts the developer, downloads and installs any updates)
  - **Pro**: Don't have to think/worry about updates, always up to date
  - **Con**: No control – bad update might crash the system

Common types of Release Management

- Continuous delivery
- Agile software development
- DevOps

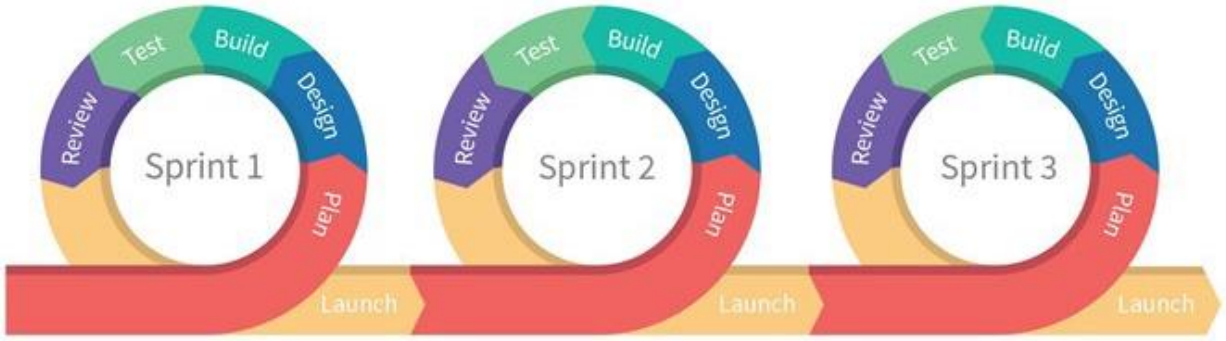
Continuous delivery

- Continuous delivery is a software engineering approach in which teams produce software in **short cycles**, ensuring that the software can be reliably released at any time.
- It aims at building, testing, and releasing software with greater speed and frequency.
- The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production.



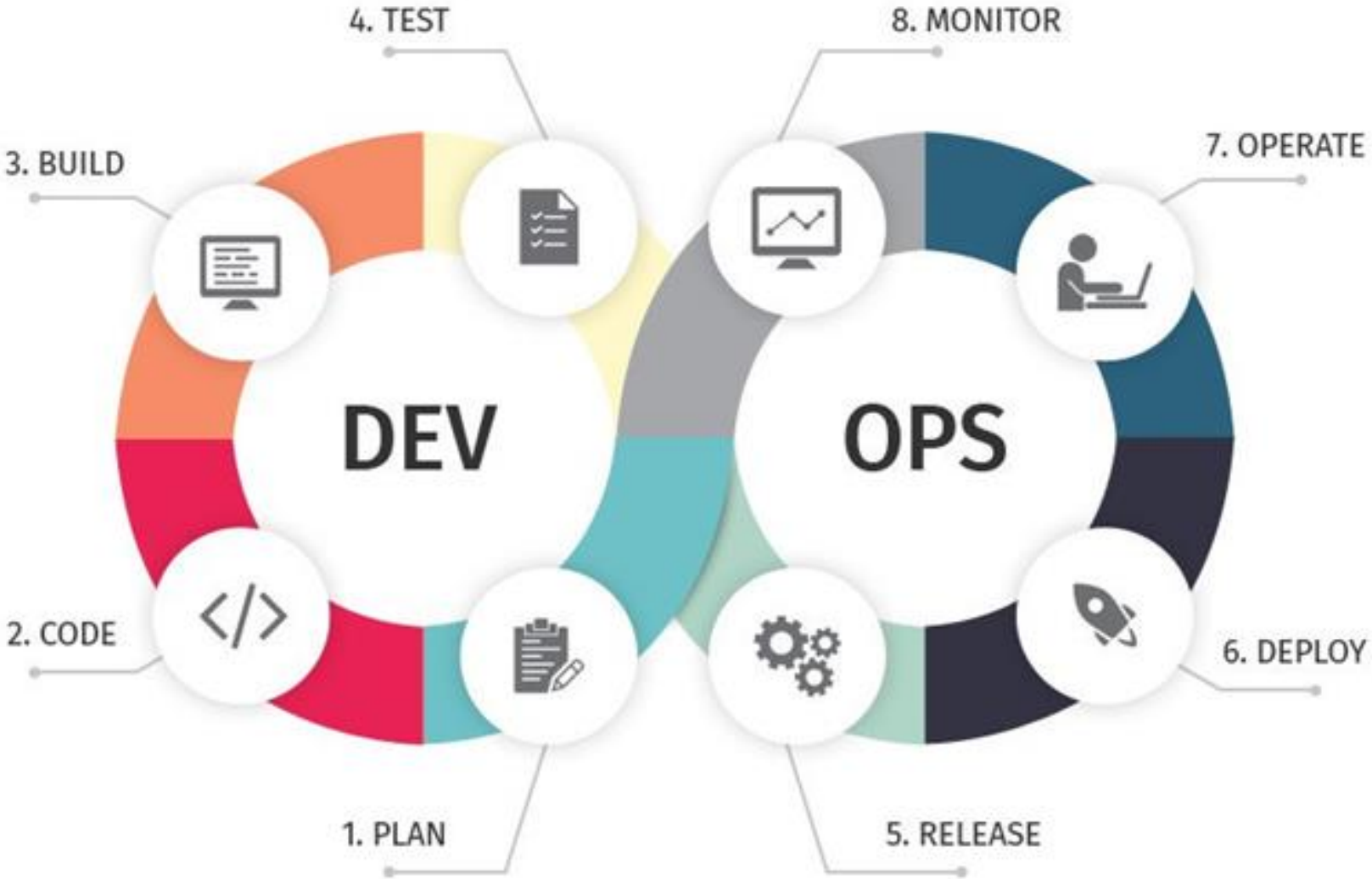
Agile development

- Agile software development describes an approach to software development under which requirements and solutions evolve through the **collaborative effort** of self-organizing and cross-functional teams and their customers/end users.
- It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.



DevOps (Development Operations)

- DevOps is a software engineering culture and practice that aims at **unifying** software development (Dev) and software operation (Ops).
- The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management.
- DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives


4. TEST
8. MONITOR
3. BUILD
7. OPERATE
DEV
OPS
2. CODE
6. DEPLOY
1. PLAN
5. RELEASE