**Lab 3.1 – Review Process Models and IDEs**

**🎯 Mục tiêu**

* Hiểu các **mô hình phát triển phần mềm (SDLC Models)** phổ biến.
* So sánh điểm mạnh – yếu của từng mô hình.
* Làm quen với **IDE** (Visual Studio, VS Code, IntelliJ, v.v.).
* Thực hành tạo **ứng dụng quản lý sản phẩm**: có chức năng thêm, sửa, xóa, xem danh sách sản phẩm.

**PHẦN 1️⃣: Báo cáo lý thuyết – *Review Process Models***

**🧠 1. Giới thiệu SDLC**

**SDLC (Software Development Life Cycle)** là quy trình chuẩn để phát triển phần mềm từ khi hình thành ý tưởng → triển khai → bảo trì.  
Nó giúp đảm bảo **chất lượng**, **thời gian**, và **chi phí**.

Các mô hình SDLC phổ biến:

1. Waterfall (Thác nước)
2. Iterative (Lặp lại)
3. Spiral (Xoắn ốc)
4. Agile (Linh hoạt)
5. DevOps (Phát triển & Vận hành kết hợp)

**📘 2. Mô tả các mô hình**

**A. Waterfall Model (Mô hình thác nước)**

**Đặc điểm:**

* Gồm các giai đoạn nối tiếp nhau:  
  Requirement → Design → Implementation → Testing → Deployment → Maintenance
* Hoàn thành 1 giai đoạn mới chuyển sang giai đoạn kế tiếp.

**Ưu điểm:**

* Dễ hiểu, rõ ràng.
* Phù hợp với dự án nhỏ, yêu cầu cố định.

**Nhược điểm:**

* Khó thay đổi yêu cầu khi đã qua giai đoạn.
* Lỗi phát hiện muộn → tốn kém sửa chữa.
* Không phù hợp dự án linh hoạt.

**B. Iterative Model (Mô hình lặp lại)**

**Đặc điểm:**

* Xây dựng phần mềm theo từng vòng lặp.
* Mỗi vòng lặp có thể gồm đầy đủ SDLC, giúp cải tiến dần sản phẩm.

**Ưu điểm:**

* Phát hiện sớm lỗi và điều chỉnh được.
* Có thể release sớm phiên bản mẫu (prototype).

**Nhược điểm:**

* Tốn thời gian quản lý nhiều vòng lặp.
* Đòi hỏi đội ngũ có kinh nghiệm.

**C. Agile Model (Mô hình linh hoạt)**

**Đặc điểm:**

* Làm việc theo **Sprint** ngắn (1–4 tuần).
* Mỗi Sprint tạo ra sản phẩm chạy được (increment).
* Giao tiếp liên tục với khách hàng.

**Ưu điểm:**

* Phản hồi nhanh, thích nghi thay đổi tốt.
* Liên tục cải tiến.
* Tăng sự hài lòng khách hàng.

**Nhược điểm:**

* Khó áp dụng nếu team không kỷ luật.
* Không phù hợp nếu khách hàng không sẵn sàng tham gia thường xuyên.

**D. Spiral Model (Mô hình xoắn ốc)**

**Đặc điểm:**

* Kết hợp giữa **Waterfall + Iterative**, tập trung vào **quản lý rủi ro**.
* Gồm 4 pha lặp lại:  
  Planning → Risk Analysis → Engineering → Evaluation

**Ưu điểm:**

* Giảm rủi ro dự án lớn/phức tạp.
* Cho phép phản hồi sớm.

**Nhược điểm:**

* Chi phí cao, yêu cầu nhân sự có kinh nghiệm cao.

**E. DevOps Model**

**Đặc điểm:**

* Kết hợp quá trình **phát triển (Dev)** và **vận hành (Ops)**.
* Tập trung vào **tự động hóa**, **CI/CD**, **triển khai liên tục**.

**Ưu điểm:**

* Rút ngắn thời gian release.
* Giảm lỗi khi triển khai.
* Liên tục giám sát, cải thiện.

**Nhược điểm:**

* Cần hạ tầng và quy trình phức tạp.
* Không phù hợp cho dự án nhỏ học tập.

**⚖️ 3. So sánh tổng quát**

| **Tiêu chí** | **Waterfall** | **Iterative** | **Agile** | **Spiral** | **DevOps** |
| --- | --- | --- | --- | --- | --- |
| **Tính linh hoạt** | Thấp | Trung bình | Cao | Cao | Rất cao |
| **Chi phí thay đổi** | Cao | Trung bình | Thấp | Trung bình | Thấp |
| **Rủi ro** | Cao | Trung bình | Thấp | Thấp | Trung bình |
| **Khách hàng tham gia** | Ít | Có | Nhiều | Nhiều | Nhiều |
| **Phù hợp với** | Dự án nhỏ, ổn định | Dự án trung bình | Dự án thay đổi liên tục | Dự án lớn, rủi ro cao | Dự án liên tục vận hành |

**📄 4. Kết luận**

* Nếu bạn làm **bài lab học tập** → chọn **Agile** (vì dễ demo, dễ chia sprint).
* Nếu làm **phần mềm doanh nghiệp cố định yêu cầu** → chọn **Waterfall**.
* Nếu muốn học **CI/CD và DevOps pipelines** → chọn **DevOps**.

**PHẦN 2️⃣: Thực hành – *Tạo Project “Product Management”***

**🎯 Mục tiêu**

Xây dựng ứng dụng CRUD sản phẩm:

* Thêm sản phẩm
* Xem danh sách sản phẩm
* Sửa sản phẩm
* Xóa sản phẩm

**💻 Công cụ đề xuất**

| **IDE** | **Ngôn ngữ** | **Gợi ý Framework** |
| --- | --- | --- |
| VS Code | JavaScript / TypeScript | React / Next.js / Node.js |
| Visual Studio | C# | ASP.NET MVC / Blazor |
| IntelliJ IDEA | Java | Spring Boot |
| PyCharm | Python | Django / Flask |

**🚀 Ví dụ thực hành bằng JavaScript (Node.js + Express)**

**1️⃣ Cấu trúc thư mục**

product-management/

├── package.json

├── server.js

├── routes/

│ └── products.js

└── data/

└── products.json

**2️⃣ Cài đặt**

npm init -y

npm install express body-parser cors

**3️⃣ Tạo file server.js**

const express = require('express');

const bodyParser = require('body-parser');

const cors = require('cors');

const productRoutes = require('./routes/products');

const app = express();

app.use(cors());

app.use(bodyParser.json());

app.use('/api/products', productRoutes);

const PORT = 3000;

app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

**4️⃣ File routes/products.js**

const express = require('express');

const router = express.Router();

let products = [

{ id: 1, name: "Laptop", price: 1500 },

{ id: 2, name: "Mouse", price: 25 },

];

// GET all

router.get('/', (req, res) => res.json(products));

// POST add

router.post('/', (req, res) => {

const newProduct = { id: Date.now(), ...req.body };

products.push(newProduct);

res.json(newProduct);

});

// PUT update

router.put('/:id', (req, res) => {

const id = parseInt(req.params.id);

const index = products.findIndex(p => p.id === id);

if (index !== -1) {

products[index] = { ...products[index], ...req.body };

res.json(products[index]);

} else res.status(404).json({ message: "Not found" });

});

// DELETE

router.delete('/:id', (req, res) => {

const id = parseInt(req.params.id);

products = products.filter(p => p.id !== id);

res.json({ message: "Deleted" });

});

module.exports = router;

**🧩 Kết quả mong đợi**

Khi chạy:

node server.js

API sẽ chạy ở http://localhost:3000/api/products

* GET /api/products → xem danh sách
* POST /api/products → thêm
* PUT /api/products/:id → sửa
* DELETE /api/products/:id → xóa

**📄 5. Báo cáo thực hành (mẫu nộp)**

**Tên sinh viên:** Nguyễn Văn A  
**Lớp:** SE1234

**Nội dung:**

* Tìm hiểu và so sánh các mô hình SDLC (Waterfall, Agile, Spiral, DevOps).
* Ứng dụng: Xây dựng hệ thống quản lý sản phẩm CRUD bằng Node.js + Express.

**Kết quả:**

* Server chạy thành công, hiển thị danh sách sản phẩm.
* Đã thực hiện thêm, sửa, xóa qua API.

**Kết luận:**  
Qua lab này, em hiểu rõ hơn về quy trình SDLC, sự khác biệt giữa các mô hình, và thực hành áp dụng Agile model trong phát triển ứng dụng nhỏ.