



CHUẨN BỊ GÌ ĐỂ TRỞ THÀNH SOFTWARE DEVELOPER?

GV: VÕ TUẤN KIỆT
Email: kietvt@uit.edu.vn
15/05/2025

Nội Dung

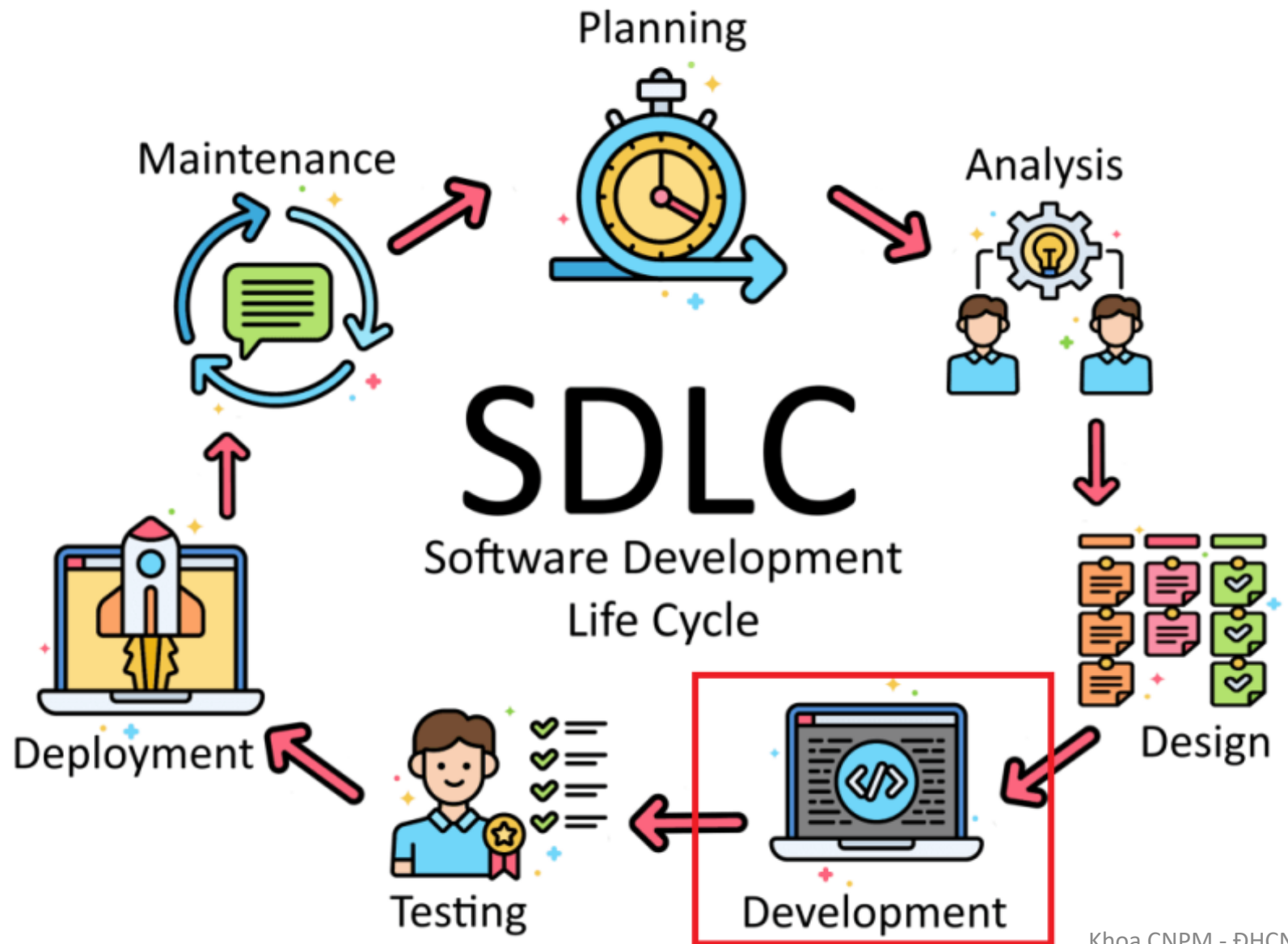


- ☐ Dev – Bạn là ai?
- ☐ Chuẩn bị gì để làm Dev?
- ☐ Những cơ hội và thách thức của AI với Dev

Dev – Bạn là ai?

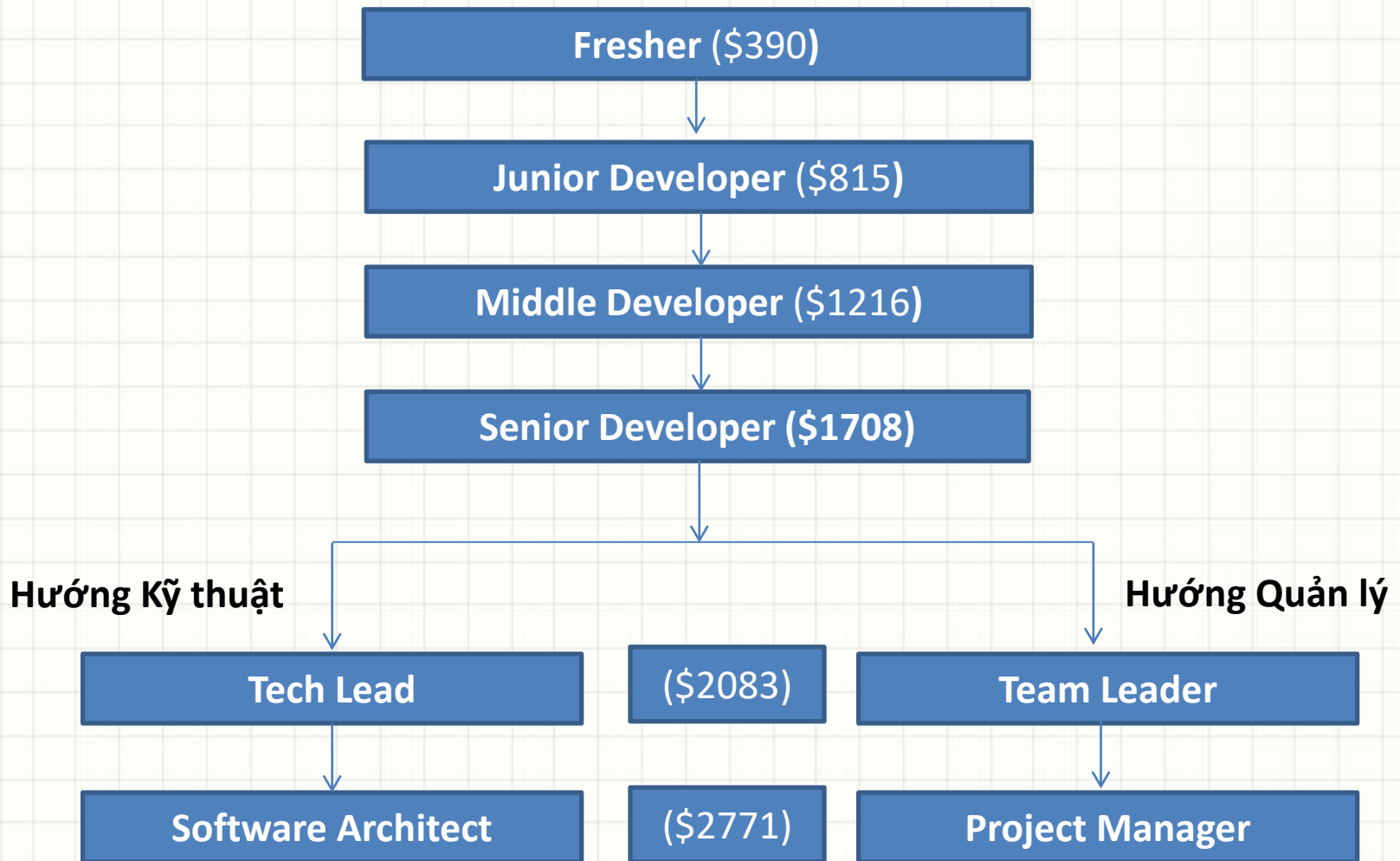


Dev – Bạn là ai?



Dev – Bạn là ai?

Mức lương gross (từ TopDev.vn report 2024-2025, Q3/2024)



Dev – Bạn là ai?

Chức Danh	Mức Độ Kinh Nghiệm	Vai Trò Chính	Trách Nhiệm
Fresher	0 – 6 tháng	Học việc, làm quen quy trình	Thực hiện các tác vụ đơn giản, được giao sát sao
Junior Developer	6 tháng – 2 năm	Hỗ trợ phát triển tính năng nhỏ	Viết code dưới sự hướng dẫn, fix bug, code review sơ bộ
Middle Developer	2 – 4 năm	Phát triển tính năng chính	Phân tích yêu cầu, thiết kế module, tối ưu code, hỗ trợ junior
Senior Developer	4 – 7 năm	Thiết kế giải pháp, mentor	Thiết kế kiến trúc module phức tạp, review code, dẫn dắt kỹ thuật
Tech Lead	5+ năm (thiên về kỹ thuật)	Dẫn dắt kỹ thuật toàn team	Định hướng công nghệ, review thiết kế, đảm bảo chất lượng kỹ thuật
Software Architect	7+ năm (thiên về kỹ thuật)	Thiết kế kiến trúc tổng thể	Đưa ra kiến trúc hệ thống, lựa chọn công nghệ, đảm bảo scalability, security
Team Leader	5+ năm (thiên về quản lý)	Quản lý team nhỏ (5-10 người)	Phân chia công việc, đánh giá hiệu suất, hỗ trợ giải quyết xung đột
Project Manager	5+ năm (thiên về quản lý)	Quản lý dự án, tiến độ	Lập kế hoạch dự án, quản lý ngân sách, giao tiếp với khách hàng

Chuẩn bị gì để làm Dev?

- ☐ Trang bị kiến thức nền tảng
- ☐ Chọn lộ trình (roadmap) cụ thể và trang bị kiến thức chuyên sâu
- ☐ Kỹ năng mềm
- ☐ Đạo đức nghề nghiệp

Kiến thức nền tảng

Kỹ năng lập trình cơ bản

- ✓ Cấu trúc dữ liệu & Giải thuật
- ✓ Tư duy logic & giải quyết vấn đề
- ✓ Thuần thục ít nhất 1 ngôn ngữ lập trình phổ biến (Python, Java, C++)

Nguyên lý về Kỹ thuật phần mềm

- ✓ Hiểu về SDLC, Agile, Waterfall, DevOps
- ✓ Kiến thức về viết code sạch (Clean Code), thiết kế phần mềm hướng đối tượng (SOLID principles)

Cơ sở dữ liệu (Databases)

- ✓ Kiến thức cơ bản về RDBMS (MySQL, PostgreSQL) và NoSQL (MongoDB)
- ✓ Kiến thức cơ bản về lưu trữ trên Cloud
- ✓ Biết cách thiết kế cơ sở dữ liệu và viết các truy vấn SQL

Kiến thức nền tảng (tt)

Kiến thức về Hệ điều hành & Mạng máy tính

- ✓ Các khái niệm cơ bản về process, memory, file systems
- ✓ Hiểu về HTTP/HTTPS, TCP/IP, REST API

Công cụ & Môi trường phát triển phần mềm

- ✓ Sử dụng thành thạo Git/GitHub
- ✓ Làm việc với IDE/Editor (VSCode, IntelliJ, Eclipse)
- ✓ Kiến thức cơ bản về DevOps (Docker, CI/CD, Kubernetes)

Kiến thức chuyên sâu theo roadmap

Chọn roadmap nào? (tham khảo thêm về những roadmap từ <https://roadmap.sh>)

- ☐ Web Development 
- ☐ Mobile Development
- ☐ Game Development
- ☐ Block chain Development
- ☐ Embedded Systems Development
- ☐ Security Development
- ☐

Kiến thức chuyên sâu theo roadmap

Different Types of Web Development



**Front End
Development**



**Back-End
Development**



**Full Stack
Development**

Kiến thức chuyên sâu theo roadmap

Front-end Developer

- HTML
- CSS
- Javascript, PHP
- Frameworks:
 - ☐ Bootstrap
 - ☐ jQuery
 - ☐ React
 - ☐ Angular
 - ☐ Vue.js
 - ☐ Typescript.js

Kiến thức chuyên sâu theo roadmap

Back-end Developer

Frameworks:

- ☐ Python/Flask, Python/Django
- ☐ Javascript, C#/ASP.NET Core
- ☐ PHP/Laravel, PHP/CodeIgniter, PHP/CakePHP
- ☐ Ruby/Ruby on Rails
- ☐ Java/Spring Boot, Java/Hibernate, Java/Struts
- ☐ Node.js-based frameworks: Express.js, Koa.js, Meteor.js, Nest.js

Hiểu về RESTful API, GraphQL và các phương pháp xác thực:

- ☐ Username/Password
- ☐ Token-based authentication (JWT)
- ☐ OAuth2 (third-party authentication)
- ☐ Multi-Factor authentication (MFA)
- ☐ Biometric authentication

Kỹ năng mềm

1. Kỹ Năng Giao Tiếp

- Diễn đạt rõ ràng: Giải thích vấn đề kỹ thuật cho người không chuyên (PM, khách hàng) dễ hiểu.
- Lắng nghe chủ động: Hiểu đúng yêu cầu của đồng đội và khách hàng.
- Giao tiếp qua viết: Viết email, tài liệu kỹ thuật, báo cáo rõ ràng, súc tích.
- Trình bày & thuyết trình: Trình bày ý tưởng kỹ thuật mạch lạc, thuyết phục.

2. Kỹ Năng Làm Việc Nhóm

- Biết làm việc chung với các team khác như QA, DevOps, Product Owner.
- Tôn trọng ý kiến đồng đội, sẵn sàng hỗ trợ và chia sẻ kiến thức.
- Giải quyết xung đột nội bộ một cách chuyên nghiệp.

3. Kỹ Năng Giải Quyết Vấn Đề

- Phân tích vấn đề rõ ràng, tìm ra giải pháp khả thi, không đổ lỗi.
- Đánh giá tác động của các giải pháp và chọn hướng đi tối ưu.

4. Kỹ Năng Tư Duy Phản Biện

- Luôn đặt câu hỏi "Tại sao?" và "Có cách nào tốt hơn không?".
- Đánh giá rủi ro, hậu quả khi áp dụng một công nghệ hoặc kiến trúc.

Kỹ năng mềm (tt)

5. Khả Năng Tự Học và Thích Ứng

- Luôn cập nhật công nghệ mới, xu hướng phát triển phần mềm.
- Linh hoạt thích ứng khi thay đổi môi trường làm việc, công nghệ, dự án.

6. Kỹ Năng Quản Lý Thời Gian

- Biết ưu tiên công việc quan trọng
- Tránh sa đà vào chi tiết kỹ thuật không cần thiết

7. Kỹ Năng Đàm Phán và Thuyết Phục

- Đàm phán deadline hợp lý với PM/khách hàng.
- Thuyết phục team chấp nhận giải pháp kỹ thuật của mình dựa trên lập luận vững chắc.

8. Khả Năng Chịu Áp Lực

- Biết cách cân bằng công việc và cuộc sống
- Giữ bình tĩnh khi gặp sự cố hệ thống hoặc các tình huống khẩn cấp.

9. Tư Duy Cầu Tiến

- Nhận thức được điểm yếu của bản thân và luôn nỗ lực cải thiện.
- Xem thất bại là bài học, không ngại thử thách mới.

Đạo đức nghề nghiệp

- 1. Xử lý dữ liệu cá nhân và bảo mật thông tin:** Không tự ý truy cập, tiết lộ hay sử dụng thông tin người dùng trái phép.
- 2. Trung thực về năng lực và tiến độ:** Không "vẽ hươu vẽ vượn" về sản phẩm, công nghệ mà bạn không nắm vững.
- 3. Chống lại các hành vi phi đạo đức trong phát triển sản phẩm:** Ví dụ, không phát triển phần mềm chứa mã độc, phần mềm đánh cắp thông tin, hoặc hỗ trợ các hành vi gian lận.
- 4. Tuân thủ luật bản quyền:** Không sử dụng phần mềm hoặc thư viện lậu, tôn trọng giấy phép phần mềm (GPL, MIT, Apache...).
- 5. Tôn trọng đồng nghiệp:** Không "chiếm công" người khác, không "dìm" đồng nghiệp để nổi bật.

Cơ hội và thách thức của AI đối với Dev

Cơ hội:

- ✓ Tăng năng suất làm việc
- ✓ Nhiều cơ hội việc làm mới
- ✓ Tạo nên các sản phẩm mới thông minh hơn

Thách thức:



- ✓ Dễ bị thay thế nếu thiếu tư duy giải quyết vấn đề
- ✓ Gia tăng sự cạnh tranh toàn cầu
- ✓ Công nghệ thay đổi liên tục
- ✓ Rủi ro pháp lý, bảo mật

Một Số Website Hữu Ích

- 1) <https://topdev.vn>
- 2) <https://roadmap.sh>
- 3) <https://dev.to>
- 4) <https://leetcode.com>
- 5) <https://www.hackerrank.com>
- 6) <https://www.coursera.org>
- 7) <https://www.pluralsight.com>
- 8) <https://www.udemy.com>

Tham gia các dự án mã nguồn mở:

- **GitHub:** <https://github.com/explore>
→ Tìm các dự án đang tuyển contributor (gắn nhãn good first issue, help wanted).
- **Up For Grabs:** <https://up-for-grabs.net/>
→ Website tổng hợp các dự án đang tìm người tham gia.
- **Google Summer of Code (GSoC):** <https://summerofcode.withgoogle.com/>
→ Dành cho sinh viên, vừa học vừa được tài trợ tiền (hàng nghìn USD).
- **Outreachy:** <https://www.outreachy.org/>
→ Hỗ trợ các bạn muốn tham gia open source và có ngân sách hỗ trợ.

Tham gia các nền tảng freelance (kiếm tiền):

- **Upwork:** <https://www.upwork.com/>
→ Nhiều dự án lập trình, mức thu nhập cao nhưng cạnh tranh lớn.
- **Freelancer:** <https://www.freelancer.com/>
→ Có thể bắt đầu từ các dự án nhỏ, tăng dần uy tín.
- **Fiverr:** <https://www.fiverr.com/>
→ Đăng dịch vụ cố định (VD: “Tôi sẽ làm web bằng React chỉ với 100\$”).
- **TopTal:** <https://www.toptal.com/>
→ Yêu cầu kỹ năng cao, nhưng bù lại dự án chất lượng và thu nhập tốt.