

ĐỀ CƯƠNG MÔN HỌC

CTT537 – Đồ Án Công Nghệ Phần Mềm

1. THÔNG TIN CHUNG

(Hướng dẫn: mô tả các thông tin cơ bản của môn học)

Tên môn học (tiếng Việt):	Đồ án công nghệ phần mềm
Tên môn học (tiếng Anh):	Software Engineering Capstone
Mã số môn học:	CTT537
Thuộc khối kiến thức:	Chuyên ngành
Số tín chỉ:	4
Số tiết lý thuyết:	45
Số tiết thực hành:	30
Số tiết tự học:	90
Các môn học tiên quyết	Nhập môn công nghệ phần mềm Phương pháp lập trình hướng đối tượng Lập trình Windows Phát triển ứng dụng web Phát triển game Phát triển phần mềm cho thiết bị di động Phân tích và quản lý yêu cầu phần mềm Phân tích và thiết kế phần mềm Kiểm chứng phần mềm Quản lý dự án phần mềm

2. MÔ TẢ MÔN HỌC (COURSE DESCRIPTION)

(Hướng dẫn: một đoạn văn mô tả tóm tắt về nội dung của môn học)

Lượng sinh viên tốt nghiệp với điểm số cao nhưng không tìm được việc làm ngày càng nhiều. Một trong các nguyên nhân là do sinh viên thiếu các kinh nghiệm làm việc thực tế, vận dụng các kiến thức và kỹ năng đã được học một cách rời rạc, chưa có sự *tổng hợp*, để giải quyết *vấn đề thực tế*. Ngoài ra,

các sinh viên cũng gặp khó khăn khi tìm hiểu các kiến thức về lĩnh vực (domain knowledge), hay làm quen với một công cụ, kỹ thuật, hay phương pháp mới, cho dự án thực tế.

Môn học đồ án công nghệ phần mềm *tích hợp tất cả các kiến thức và kỹ năng* sinh viên đã được học, nhằm mục đích giúp sinh viên *làm việc theo nhóm*, thực hiện việc xác định vấn đề, phân tích và quản lý yêu cầu, thiết kế hệ thống, hiện thực hóa, kiểm thử, đảm bảo chất lượng, quản trị dự án, tài liệu hóa và chuyển giao một hệ thống phần mềm *tương đương với các phần mềm thương mại thực tế*.

Điểm khác biệt quan trọng ở môn học này là *vấn đề cần giải quyết là thực tế*, giải pháp do sinh viên *tự trải nghiệm và phát hiện*, dựa trên tất cả các kiến thức đã được học, và *kiến thức tự học bổ sung* nhằm hỗ trợ giải quyết vấn đề khi các kiến thức đã có chưa đủ để thực hiện giải pháp, tương tự như đang làm việc thực sự tại các công ty, tổ chức. Bản thân mỗi thành viên trong nhóm có thể đảm nhiệm vai trò nhà phát triển toàn phần (*Full Stack Developer*).

Các nội dung trọng tâm bao gồm việc *áp dụng* các phương pháp xác định vấn đề, xây dựng bản mẫu, tạo bằng chứng khái niệm, phân tích yêu cầu, thiết kế phần mềm, hiện thực hóa phần mềm, kiểm thử phần mềm; *áp dụng* các mô hình sản xuất phần mềm (Kanban, Scrum, XP, RUP, thác nước); *áp dụng* các phương pháp quản lý dự án (ước lượng, tạo lịch trình, kế hoạch chuyển giao); *đề xuất* các độ đo chất lượng; *thử nghiệm* và áp dụng các công cụ và công nghệ hỗ trợ.

Ngoài ra, một số kiến thức nền và chủ đề kỹ thuật thông dụng trong ngành công nghiệp phần mềm, *liên quan trực tiếp đến việc phát triển đồ án, cũng có thể được giới thiệu* nhằm giúp sinh viên trau dồi khả năng tự tìm hiểu, nghiên cứu, ví dụ như phương pháp học một lĩnh vực mới, một ngôn ngữ lập trình mới, một khung ứng dụng mới, một kiến trúc phần mềm mới (kiến trúc vi dịch vụ – microservice architecture, nguồn sự kiện – event sourcing, kỹ thuật tách biệt truy vấn và mệnh lệnh – CQRS, kiến trúc khung ứng dụng), phương pháp thiết kế định hướng từ nghiệp vụ (domain-driven design), phương pháp phát triển định hướng từ kiểm thử (test-driven development), phương pháp đọc mã nguồn hệ thống phức tạp, kiểm thử tự động, tối ưu hóa hệ thống, tái cấu trúc hệ thống, bảo mật phần mềm, hiệu chỉnh phần mềm, tùy biến phần mềm, phát triển trò chơi, ứng dụng học máy, ứng dụng xử lý ngôn ngữ tự nhiên, ứng dụng thị giác máy tính, ứng dụng blockchain.

3. MỤC TIÊU MÔN HỌC (COURSE GOALS)

(Hướng dẫn: Liệt kê các mục tiêu môn học, từ 5-8 mục tiêu ở mức độ tổng quát. Sử dụng động từ Bloom ở mức độ nhóm. Mỗi mục tiêu môn học được mapping với chuẩn đầu ra cấp chương trình)

Sinh viên học xong môn học này có khả năng :

Mục tiêu	Mô tả (mức tổng quát)	CĐR CDIO của chương trình
----------	------------------------	---------------------------

G1	Xác định được vấn đề thực tế, phân tích được yêu cầu nghiệp vụ, đề xuất được giải pháp bằng phần mềm và phân tích được các ảnh hưởng của giải pháp đối với các cá nhân, tổ chức và xã hội.	2.1.1, 2.1.2, 2.1.3 5.2.1, 5.2.2, 5.2.3
G2	Lựa chọn và tài liệu hóa được quy trình thực hiện, lên được kế hoạch thực hiện dự án.	2.5.1; 2.5.2; 2.5.3
G3	Phân tích, thiết kế, hiện thực hóa, kiểm thử, triển khai, vận hành và tài liệu hóa được giải pháp cho một vấn đề thực tế trong ngành kỹ nghệ phần mềm.	5.1.1, 5.1.2, 5.1.3 5.3.1, 5.3.2, 5.3.3 5.4.1, 5.4.2, 5.4.3 6.2.1, 6.2.2, 6.2.3 6.3.1, 6.3.2
G4	Ước lượng và đánh giá được kết quả của giải pháp đề xuất. So sánh được giải pháp đề xuất với các hướng thay thế khác.	6.1.1, 6.1.2, 6.1.3
G5	Tự tìm hiểu và sử dụng được các công cụ, ngôn ngữ, công nghệ, phương pháp để xây dựng giải pháp.	2.4.1, 2.4.2, 2.4.3
G6	Làm việc nhóm để xây dựng được phần mềm kích cỡ khoảng 75.000 dòng mã nguồn theo một mô hình lựa chọn.	4.2.1; 4.2.2; 4.2.3 4.3.2 4.4.3
G7	Nhận thức được trách nhiệm và đạo đức của mình khi thực hiện một dự án phần mềm.	3.3.2, 3.3.3

4. CHUẨN ĐẦU RA MÔN HỌC

(Hướng dẫn: Mô tả chi tiết các chuẩn đầu ra của môn học. Ứng với mỗi mục tiêu ở mục phía trên có thể có 1 hay nhiều chuẩn đầu ra chi tiết. Đánh mã số chuẩn đầu ra môn học ở cấp 2 tương ứng với mỗi mục tiêu môn học. Mức độ được thể hiện bằng các ký hiệu I-Introduce, T-Teach và U-Utilize. Các động từ mô tả được sử dụng từ các động từ chi tiết của Bloom cho mức độ tương ứng – xem thêm bảng các động từ Bloom chi tiết cho ngành kỹ thuật.)

Chuẩn đầu ra	Mô tả (Mức chi tiết - hành động)	Mức độ (I/T/U)
G1.1	Xác định được vấn đề thực tế và/hoặc mối quan tâm trong các hoạt động của con người, đánh giá khó khăn, thử thách trong việc thực hiện dự án.	T, U
G1.2	Phân tích được vấn đề, yêu cầu nghiệp vụ và giải pháp phần	T, U

	mềm trong sự liên quan đến các cá nhân, tổ chức và xã hội.	
G1.3	Liệt kê được giải pháp tổng quát khả dĩ và các bước hiện thực hóa.	T, U
G2.1	Lựa chọn và tài liệu hóa được quy trình thiết kế dựa trên mô hình thác nước, RUP, Scrum, hoặc Kanban, lên được kế hoạch thực hiện dự án.	T, U
G3.1	Phân tích, thiết kế, hiện thực hóa và kiểm thử giải pháp một cách hoàn chỉnh, tài liệu hóa được các tài liệu dự án yêu cầu.	T, U
G4.1	Ước lượng và đánh giá được giải pháp đề xuất. So sánh được giải pháp đề xuất với các hướng thay thế khác.	T, U
G5.1	Lựa chọn, áp dụng được các kỹ thuật và công cụ để xây dựng giải pháp tương ứng với các hoạt động lấy yêu cầu, phân tích, thiết kế, hiện thực hóa, kiểm thử, chuyển giao, quản trị.	I, U
G5.2	Tự tìm hiểu và thuyết trình một chủ đề liên quan đến công cụ, ngôn ngữ, công nghệ, phương pháp trong lĩnh vực kỹ nghệ phần mềm được sử dụng trong dự án, trong 30 phút.	T, U
G6.1	Xây dựng được một dự án phần mềm kích cỡ khoảng 75.000 dòng mã nguồn với nhóm 10 người.	T, U
G7.1	Thảo luận các chủ đề liên quan đến nghề nghiệp, đạo đức và khía cạnh xã hội trong các hoạt động và nghề nghiệp thuộc chuyên ngành kỹ nghệ phần mềm	I, T

5. KẾ HOẠCH GIẢNG DẠY LÝ THUYẾT

(Hướng dẫn: Mô tả chi tiết quá trình giảng dạy theo từng chủ đề: tên chủ đề, danh sách các chuẩn đầu ra chi tiết tương ứng với mỗi chủ đề, các hoạt động dạy và học gợi ý, các hoạt động đánh giá nếu có)

STT	Tên chủ đề	Chuẩn đầu ra	Hoạt động dạy/ Hoạt động học (gợi ý)	Hoạt động đánh giá
1	Software Engineering Capstone Introduction	G1.1, G5.1, G7.1	Thuyết giảng Động não Vấn đáp	BTTL#1
2	<ul style="list-style-type: none"> Business Requirements <ul style="list-style-type: none"> Business Use Cases Domain Models Prototypes Proof of Concept Product Backlog 	G1.1, G1.2, G7.1, G3.1	Thuyết trình Vấn đáp Dựa vào dự án Thuyết giảng	BTTL#1 DAMH#1 DAMH#2

	<ul style="list-style-type: none"> • Build Automation • The Kanban Development Process • The Scrum Development Process • The Extreme Programming Development Process • The Rational Unified Process 			
3	<p>Enterprise Application Projects</p> <ul style="list-style-type: none"> • Microservice Architecture • Domain-Driven Design • Scalability, Event Sourcing, CQRS • New Programming Languages • Code Reading • Software Implementation • Program Optimization • Refactoring • Software Reuse, Software Frameworks • Software Security • Software Variability <p>Machine Learning Projects</p> <ul style="list-style-type: none"> • Python Language and Libraries • Linear Regression • Gradient Descent • Logistic Regression • Softmax Regression 	G3.1, G4.1, G5.1, G5.2, G6.1	<p>Thuyết trình</p> <p>Vấn đáp</p> <p>Dựa vào dự án</p> <p>Thuyết giảng</p>	<p>BTTL#2</p> <p>DAMH#2</p> <p>DAMH#3</p> <p>DAMH#4</p>

	<ul style="list-style-type: none"> • Perceptron • Neural Networks • TensorFlow • Keras • Recommender Systems 			
--	---	--	--	--

6. KẾ HOẠCH GIẢNG DẠY THỰC HÀNH (nếu có)

(Hướng dẫn: Mô tả tương tự như kế hoạch giảng dạy lý thuyết. Các chủ đề được liệt kê tuần tự và các chuẩn đầu ra, hoạt động giảng dạy và đánh giá tương ứng cho từng chủ đề.

Lưu ý: đối với hình thức thực hành là hình thức 2 – nghĩa là GVTH không lên lớp thì có thể ghi trong hoạt động dạy & học là “thảo luận và trả lời thắc mắc trên diễn đàn môn học”)

Tuần	Chủ đề	Chuẩn đầu ra	Hoạt động dạy/ Hoạt động học (gợi ý)	Hoạt động đánh giá
1	Hướng dẫn thực hiện đồ án và viết báo cáo.		Thảo luận và trả lời thắc mắc trên diễn đàn môn học	

7. ĐÁNH GIÁ

(Hướng dẫn: Mô tả các thành phần bài tập, bài thi, đồ án... dùng để đánh giá kết quả của sinh viên khi tham gia môn học này. Bên cạnh mỗi nhóm bài tập, bài thi... cần có tỉ lệ % điểm tương ứng)

Mã	Tên	Mô tả (gợi ý)	Các chuẩn đầu ra được đánh giá	Tỉ lệ (%)
BTTL	Bài tập tại lớp			30%
BTTL#1	Thuyết trình vấn đề đồ án giải quyết và tổng quan về giải pháp.	<ul style="list-style-type: none"> - Business Use Cases. - Domain Models. - Prototypes. - Proof of Concept. - Product Backlog. 	2.1.1, 2.1.2, 2.1.3 5.2.1, 5.2.2, 5.2.3	10%
BTTL#2	Thuyết trình giải pháp kỹ thuật cho đồ án đề xuất.	<ul style="list-style-type: none"> - Hệ thống quản lý cấu hình và build tự động. - Ngôn ngữ lập trình, công cụ, thư viện cho đồ án. 	2.1.1, 2.1.2, 2.1.3 5.2.1, 5.2.2, 5.2.3	20%

		- Giải pháp kỹ thuật cụ thể cho bài toán đặc thù.		
DAMH	Đề án môn học			40%
DAMH#1	<p>Đề xuất ý tưởng cho đề án.</p> <p>Xác định vấn đề, yêu cầu, giải pháp, tính khả thi của đề án.</p> <p>Chuẩn bị tài nguyên, công cụ cần thiết để thực hiện đề án.</p>	<ul style="list-style-type: none"> - Executive Summary. - Business Use Cases. - Domain Models. - Project Vision. - Feature Tree. - Process Flow. - Prototypes. - Proof of Concept. - Product Backlog. - Risk List. - Software Configuration Systems. - Build Automation System. 	<p>2.4.1, 2.4.2, 2.4.3</p> <p>2.1.1, 2.1.2, 2.1.3</p> <p>5.2.1, 5.2.2, 5.2.3</p>	10%
DAMH#2	<p>Xác định mô hình để thực hiện đề án.</p> <p>Xác định giải pháp cho đề án.</p> <p>Bổ sung kiến thức còn thiếu hụt.</p>	<ul style="list-style-type: none"> - Software Process Definition. - Software Architecture and Design. - Mã nguồn các ví dụ minh họa cho các kiến thức thiếu hụt. 	<p>2.1.1, 2.1.2, 2.1.3</p> <p>5.2.1, 5.2.2, 5.2.3</p> <p>5.3.1, 5.3.2, 5.3.3</p>	10%
DAMH#3	<p>Chi tiết hóa giải pháp.</p> <p>Hiện thực hóa phần mềm.</p> <p>Kiểm thử, đo lường.</p> <p>So sánh với các hệ thống tương tự.</p>	Các kết quả tương ứng với quy trình và mục tiêu đề ra trong 5 tuần tiếp theo của môn học.	<p>5.1.1, 5.1.2, 5.1.3</p> <p>5.3.1, 5.3.2, 5.3.3</p> <p>5.4.1, 5.4.2, 5.4.3</p> <p>6.1.1, 6.1.2, 6.1.3</p> <p>6.2.1, 6.2.2, 6.2.3</p>	10%

			6.3.1, 6.3.2	
DAMH#4	Hoàn thiện đồ án và chuyển giao kết quả.	Các kết quả tương ứng với quy trình và mục tiêu đề ra trong 5 tuần cuối của môn học.	Tất cả các mục tiêu	10%
LTCK	Thi lý thuyết cuối kỳ	<ul style="list-style-type: none"> - Câu trả lời cho câu hỏi vấn đáp. - Sự giải thích cho các sản phẩm đã hoàn thành của đồ án. 	Tất cả các mục tiêu	30%

8. TÀI NGUYÊN MÔN HỌC

Giáo trình

Craig Larman (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Third Edition. (thư viện ĐHQG TPHCM).

Jonathan Rasmusson (2010). *The Agile Samurai: How Agile Masters Deliver Great Software*. Pragmatic Bookshelf.

Eric Brechner (2015). *Agile Project Management with Kanban*. Microsoft Press.

Wes McKinney (2017). *Python for Data Analysis. Data Wrangling with Pandas, NumPy, and IPython*. 2nd Edition. O'Reilly.

Aurelien Geron (2019). *Hands on Machine Learning with Scikit Learn and TensorFlow*. O'Reilly Media.

Francois Chollet (2018). *Deep Learning with Python*. Manning Publications Co.

Tài liệu tham khảo

Kenneth S. Rubin (2012). *Essential Scrum A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional.

Mike Cohn (2004). *User Stories Applied For Agile Software Development*. Addison-Wesley.

Jeff Patton and Peter Economy (2014). *User Story Mapping: Discover the Whole Story, Build the Right Product*. O'Reilly Media.

Steve McConnell (2004). *Code Complete*. Second Edition. (thư viện CTTT)

Martin Fowler et al. (1999). *Refactoring Improving The Design Of Existing Code*. (thư viện CTTT)

Kent Beck (2004). *Test-Driven Development By Example*. (thư viện CTTT)

Leszek A. Maciaszek (2001). *Requirements Analysis and System Design: Developing Information Systems with UML*. Addison-Wesley. (thư viện trường KHTN TPHCM)

Perry William (2006). *Effective Methods for Software Testing*. Wiley Publishing, Inc. (thư viện trường KHTN TPHCM)

David C. Hay (2002). *Requirements Analysis - From Business Views to Architecture*.

Martin Fowler (1997). *Analysis Patterns - Reusable Object Models*.

Eric Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley. 2003.

Martin C. Robert and Martin Micah (2006). *Agile Principles, Patterns, and Practices in C#*.

Xin Chen (2004). *Developing Application Frameworks in .NET*. Apress.

Roger S Pressman (2001). *Software engineering: A Practitioner's Approach*. Palgrave Macmillan. (thư viện ĐHQG TPHCM).

Ian Sommerville (2004). *Software Engineering*. Addison Wesley. (thư viện ĐHQG TPHCM).

Machine Learning (Coursera): <https://www.coursera.org/learn/machine-learning/home/welcome>

Neural Networks and Deep Learning (Coursera): <https://www.coursera.org/learn/neural-networks-deep-learning/home/welcome>

Tài nguyên khác

GIT/SVN, Xcode, Android Studio, Visual Studio, NodeJS, Python, Tensorflow, Keras

9. CÁC QUY ĐỊNH CHUNG

- Sinh viên cần tuân thủ nghiêm túc các nội quy và quy định của Khoa và Trường.
- Sinh viên không được vắng quá 3 buổi trên tổng số các buổi học lý thuyết.
- Đối với bất kỳ sự gian lận nào trong quá trình làm bài tập hay bài thi, sinh viên phải chịu mọi hình thức kỷ luật của Khoa/Trường.